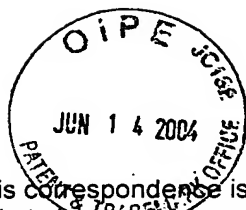


RWS-S3065



I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

By: \_\_\_\_\_

Date: 6/10/04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applic. No. : 10/825,755 Confirmation No: N/A  
Applicant : Lorenzo Di Gregorio  
Filed : April 15, 2004  
Art Unit : N/A  
Examiner : N/A  
  
Docket No. : RSW-S3065  
Customer No.: 24131

**CLAIM FOR PRIORITY**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Claim is hereby made for a right of priority under Title 35, U.S. Code, Section 199, based upon the German Patent Application 103 17 370.6, filed April 15, 2003.

A certified copy of the above-mentioned foreign patent application is being submitted herewith.

Respectfully submitted,

For Applicant

Date: \_\_\_\_\_

Gregory L. Mayback  
Reg. No. 40,719

Lerner and Greenberg, P.A.  
Post Office Box 2480  
Hollywood, FL 33022-2480  
Tel: (954) 925-1100  
Fax: (954) 925-1101

/tk



## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 103 17 370.6  
**Anmeldetag:** 15. April 2003  
**Anmelder/Inhaber:** Infineon Technologies AG,  
81669 München/DE  
**Bezeichnung:** Scheduler zum Melden einer Ablaufzeit  
**IPC:** G 06 F, H 04 L

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 29. April 2004  
**Deutsches Patent- und Markenamt**  
Der Präsident  
Im Auftrag

Agurks

## Scheduler zum Melden einer Ablaufzeit

Die Erfindung betrifft einen Scheduler für ein Speichersystem, der einer Datenverarbeitungseinheit meldet, dass in dem Speichersystem gespeicherte Daten für einen von dem Benutzer definierten Zeitraum abgespeichert sind.

Figur 1 zeigt prinzipiell den Aufbau bei einer Datenverarbeitung. Von einer Datenquelle werden parallel oder seriell Daten an eine Datenverarbeitungseinheit abgegeben, die die Daten entsprechend einer Berechnungsvorschrift verarbeitet und die verarbeiteten Daten an eine nachgeschaltete Datensenke abgibt.

Am Dateneingang der Datenverarbeitungseinheit kommt der Datenstrom entsprechend einer Datenankunftskurve an. In Figur 2 ist eine lineare bzw. ideale Datenankunftskurve dargestellt. Bei einer realen Datenübertragung ist die Datenankunftskurve nicht linear, sondern weist verschiedene Steigungen bzw. Sprünge auf, die beispielsweise von Datenbursts stammen.

Die Datenverarbeitungseinheit verarbeitet die Daten entsprechend einer Datenbedienkurve, die in Figur 2 dargestellt ist. Kommen, wie bei dem in Figur 2 dargestellten Beispiel, mehr Daten am Eingang E der Datenverarbeitungseinheit an, als durch die Datenverarbeitungseinheit verarbeitet werden können, ist ein Zwischenspeicher am Eingang E der Datenverarbeitungseinheit vorzusehen, um die überschüssigen Daten zwischenspeichern. Die minimale Speichergröße des Zwischenspeichers entspricht der maximalen Differenz zwischen der Ankunftskurve und der Bedienkurve. Die in der Datenverarbeitungseinheit verarbeiteten Daten sind der nachgeschalteten Datensenke innerhalb einer bestimmten Verzögerungszeit  $\Delta T$  zur Verfügung zu stellen. Der in der Datenverarbeitungseinheit ankommende Datenstrom ist nach einer vorbestimmten Verzögerungszeit  $\Delta T$  an die nachgeschaltete Datensenke als abzuholende Daten zu melden. In dem Zeitraum zwischen der An-

kunftszeit der Daten und der Abgabezeit, wenn die Zeitverzögerung den gewünschten Zeitverzögerungswert  $\Delta T$  erreicht, bleiben die Daten zur Verfügung. Nach Ablauf dieser Zeitspanne bzw. Zeitdauer werden die nicht an die Datensenke weitergeleiteten bzw. verarbeiteten Daten zwingend sofort gesendet, umgespeichert oder gelöscht. Die Datenquelle erzeugt den Datenstrom (traffic) mit einer abschnittsweise nahezu linearen Ankunftscurve (Arrival Curve) am Ausgang der Datenquelle. Der Datenstrom wird durch den Datenübertragungskanal modifiziert. Die Datenverarbeitungseinheit gibt die empfangenen Daten nach der vorgegebenen Zeitverzögerung  $\Delta T$  ab. Es ist dabei erstrebenswert, dass die lineare Ankunftscurve der Datenquelle mit einer vorbestimmten Zeitverzögerung am Ausgang A der Datenverarbeitungseinheit rekonstruiert wird. Die von der Datenverarbeitungseinheit abgegebenen Daten des Datenstromes können an verschiedene Datensenken geroutet werden. Bei einer praktischen Anwendung innerhalb eines Speicherverwaltungssystems ist es notwendig, um Verklemmungs- bzw. Dead-Lock-Situationen zu vermeiden, eine Ablaufzeit (Time Out) für die ankommenden Daten zu implementieren. Nach dem Ablauf der Ablaufzeit  $\Delta T$  sind die zwischengespeicherten Daten entweder zu löschen oder in einen anderen Speicher abzuspeichern, um den Speicherplatz des Pufferspeichers freizugeben. Zur Erfüllung von Quality of Service-Anforderungen ist es dabei von Bedeutung, eine deterministische vorbestimmte genaue Zeitverzögerung  $\Delta T$  einzuhalten.

Bei dem sog. Time Stamping nach dem Stand der Technik wird jedes ankommende Datenpaket DP des am Eingang E der Datenverarbeitungseinheit ankommenden Datenstroms mit einem Zeitstempel (Time Stamp) versehen, die die Ankunftszeit am Eingang E angibt. Das ankommende Datenpaket DP wird eingelesen und der Zeitpunkt der Ankunft des Datenpakets wird gemessen. Die gemessene Datenankunftszeit wird als Zeitstempel in einer Tabelle abgespeichert. Anschließend wird periodisch kontrolliert, ob das empfangene Datenpaket sich bereits länger als die vorgegebene Verzögerungszeit  $\Delta T$  in der Datenverarbei-

tungseinheit befindet. Anschließend werden diejenigen Datenpakete gelöscht oder weggespeichert, deren Ablaufzeit erreicht ist.

5 Ein gravierender Nachteil beim Time Stamping besteht darin, dass ein Datenstrom, der aus einer Vielzahl von Datenpaketen besteht, mit einer entsprechenden Anzahl von Zeitstempeln bzw. Time Stamps durch die Datenverarbeitungseinheit verwaltet werden muss. Es muss einerseits für jeden Zeitstempel ein  
10 zugehöriger Speicherplatz zur Verfügung gestellt werden und darüber hinaus sind sehr viele Zeitvergleiche notwendig, so dass der schaltungstechnische Aufwand innerhalb der Datenverarbeitungseinheit erheblich zunimmt, wenn die Vergleiche der  
15 verschiedenen Zeitstempel gleichzeitig erfolgen sollen. Werden die Zeitvergleiche aufgrund der Vielzahl der Zeitstempel sequentiell durchgeführt, nimmt die Ungenauigkeit bzgl. der Verzögerungszeit  $\Delta T$  zu.

Es wurden daher sog. ein Zeitscheiben (Time-Wheel)-  
20 Speicherverwaltungsverfahren nach dem Stand der Technik vorgeschlagen. Figur 3 zeigt eine Anordnung zur Erläuterung des Zeitscheiben-Verfahrens. Eine Datenverarbeitungseinheit empfängt von einem Scheduler die Informationen zum Abholen der zu verarbeitenden Daten aus dem Speichersystem. Der Scheduler ist über ein Netzwerk an zwei Datenquellen A, B angeschlossen. Bei dem in Figur 3 dargestellten Beispiel gibt je-  
25 de Datenquelle sequentiell 5 Datenpakete an das Netzwerk ab. Bei den Datenpaketen bzw. Datenzellen handelt es sich beispielsweise um ATM-Datenzellen mit einer vorgegebenen Datengröße von 53 Bytes. Die 5 ATM-Datenzellen werden zur Fragmentierung eines größeren Datenpakets fragmentiert. Dabei wird  
30 das große Datenpaket empfangen, in kleinere Zellen fragmentiert und die fragmentierten Zellen werden anschließend über das Netzwerk an eine Datenverarbeitungseinheit übertragen.  
35 Die Datenverarbeitungseinheit fügt die empfangenen Datenzellen zusammen, um das ursprüngliche Datenpaket wieder zu erhalten.

Die über das Netzwerk übertragenen ATM-Zellen kommen in einer beliebigen Reihenfolge bei dem Scheduler an, der sie in das Speichersystem einschreibt. Weist das Speichersystem eine re-

5 relativ große Größe  $SP_0$  auf, werden beispielsweise die Datenzellen A bis E der ersten Datenquelle A und die Datenzellen  $\alpha$  bis  $\epsilon$  der zweiten Datenquelle B, wie in Figur 3 dargestellt, in das Speichersystem in ein FIFO-Verfahren eingeschrieben.

10

Ein Problem besteht, wenn die Speichergröße des Speichersystems geringer ist, beispielsweise  $SP_1$ . In diesem Falle kann keine vollständige Kette (beispielsweise aus fünf ATM-

15 Zellen) in das Speichersystem eingeschrieben werden. Nach dem Einschreiben der ATM-Zelle  $\delta$  der zweiten Datenquelle B ist das Speichersystem voll und der Scheduler muss bei der Ankunft der nächsten ATM-Zellen  $\epsilon$ , D, E entscheiden, welche bereits abgespeicherten ATM-Zellen innerhalb des Speichersystems zu löschen sind, damit eine vollständige ATM-Zellenkette

20 in dem Speichersystem zwischengespeichert werden kann, die durch die Datenverarbeitungseinheit anschließend entsprechend verarbeitet werden kann. Der Scheduler kann daher entsprechend einer Strategie bzw. einer Policy Datenpakete zu löschen bzw. umzuspeichern, wenn das Speichersystem gefüllt ist

25 und weitere Datenpakete ankommen. Der Scheduler könnte beispielsweise das erste angekommene Datenpaket A löschen, um Speicherplatz für das nächste ankommende Datenpaket  $\epsilon$  zu schaffen. In diesem Falle wäre die vollständige Kette der ATM-Zellen der Datenquelle B in dem Speichersystem zwischengespeichert und steht für die Datenverarbeitung durch die Da-

30 tenverarbeitungseinheit zur Verfügung. Eine alternative Strategie ist das vollständige Löschen aller empfangenen Datenzellen von einer Datenquelle, beispielsweise der Datenzellen  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  der Datenquelle B, um Speicherplatz für die beiden

35 ankommenden Datenzellen D, E zu schaffen, so dass die Datenverarbeitungseinheit die Daten der Datenquelle A, d.h. die ATM-Zellen A bis E, verarbeiten kann.

Bei dem in Figur 3 dargestellten Zeitscheiben-Verfahren werden die ankommenden ATM-Zellen in der Reihenfolge ihrer Ankunft abgespeichert. Bei dem Speichersystem handelt es sich um einen FIFO-Speicher. Bei dem Zeitscheiben-Verfahren bzw. Time-Wheel-Verfahren wird die Speicherbelegung des Speichersystems als Maß für die Zeit ausgenutzt. Dazu werden die Daten herkömmlicher Weise in einen kreisförmigen bzw. zirkulären FIFO-Datenpuffer abgelegt. Wird eine ankommende Datenzelle an einer Stelle X zwischengespeichert, werden die nachfolgenden Daten bzw. Datenpakete an Speicherplätzen abgespeichert, die auf den Speicherplatz X folgen. Auf diese Weise wird der FIFO-Speicher sukzessive aufgefüllt, bis der gefüllte Speicherraum einen bestimmten Pegel erreicht bzw. der Speicher vollständig mit Daten gefüllt ist. Wenn die Menge der ankommenden Daten erfasst wird und die Daten mit einer vorgegebenen Datenrate  $R$  ankommen ist es möglich, die verstrichene Zeit zu berechnen, seit dem das Datenpaket DP bei der Position X zwischengespeichert worden ist. Da der Speicher-Voll-Pegel der Ablaufzeit der zwischengespeicherten Datenzelle bzw. dem zwischengespeicherten Datenpaket entspricht, kann das zuvor eingespeicherte Datenpaket der Datenverarbeitungseinheit durch den Scheduler als abgelaufen gemeldet werden, sobald der Speicher-Voll-Pegel erreicht ist.

Ein Nachteil des Zeitscheiben-Verfahrens bzw. Time-Wheel-Verfahrens besteht darin, dass der ankommende Datenstrom nur in seltenen Fällen eine konstante gleichbleibende Datenrate  $R$  aufweist. In vielen Anwendungsfällen bestehen zeitliche Lücken zwischen der Ankunft der verschiedenen Datenzellen bzw. Datenpaketen. Da die Datenrate zur Berechnung als konstant angenommen wird, entstehen aufgrund der zum Teil stark schwankenden Datenübertragungsraten eine sehr hohe Ungenauigkeit bei der Berechnung der Ablaufzeit.

Es wurde daher ein verbessertes Zeitscheiben-Verfahren (Enhanced-Time-Wheel) vorgeschlagen, bei dem zeitliche Lücken

zwischen der Ankunft von Datenpaketen mit sog. Dummy-Zellen bzw. Fülldatenzellen überbrückt werden. Wenn innerhalb einer einstellbaren Wartezeit kein Datenpaket bzw. keine Datenzelle an dem Scheduler ankommt, schreibt der Scheduler in den FIFO-Speicher des Speichersystems eine Fülldatenzelle ein. Figur 4 zeigt schematisch einen FIFO-Speicher, in dem eine Folge von ATM-Zellen  $Z_1, Z_2, Z_3, Z_4$  entsprechend der Reihenfolge ihrer Ankunft eingeschrieben werden. Die ATM-Zellen sind untereinander mit Zeigern bzw. Pointern verkettet und weisen eine feste Datengröße von 53 Byte auf. Durch die Verkettung erkennt die Datenverarbeitungseinheit, dass die ATM-Zellen von der gleichen Datenquelle stammen. Überschreitet die Zeitdifferenz zwischen der Ankunft einer ATM-Zelle  $Z_i$  und einer zweiten ATM-Zelle  $Z_{i+1}$  eine gewisse Wartezeit, schreibt der Scheduler eine Füllzelle in den FIFO-Speicher ein. Sobald der FIFO-Speicher voll ist, werden diejenigen ATM-Zellen gelöscht bzw. in einem anderen Speicher gespeichert, die zuerst angekommen sind und somit die ältesten ATM-Zellen darstellen. Durch das Vorsehen von Dummy- bzw. Füllzellen wird die Genauigkeit bei der Berechnung der Ablaufzeit erhöht, da die Füllzellen die Zeit wiedergeben, bei der keine ATM-Datenzellen durch den Scheduler empfangen wurden.

Ein Nachteil bei der in Figur 4 dargestellten Vorgehensweise besteht in der sog. internen Fragmentierung. Die Datenzellen weisen eine feste Datengröße auf. Handelt es sich bei den Datenzellen beispielsweise um ATM-Datenpakete bzw. ATM-Datenzellen umfassen die Datenzellen 53 Byte. Die Fülldatenzellen weisen die gleiche Größe wie die Datenzellen auf, d.h. die Fülldatenzellen umfassen ebenfalls 53 Byte. Wird bei dem Enhanced-Time-Wheel-Verfahren, wie es in Figur 4 dargestellt ist, beispielsweise ein IP-Datenpaket von 54 Byte Größe empfangen und dies in einer Protokollumwandlung in eine Folge von ATM-Datenzellen umgewandelt, erhält man eine erste Datenzelle mit 53 Byte und eine zweite ATM-Datenzelle, die nur 1 Byte Nutzdaten enthält. Die übrigen 52 Byte der zweiten ATM-Datenquelle, die keine Nutzdaten enthalten, führen ebenfalls



zu einer Belegung von Speicherplatz innerhalb des FIFO-Speichers und somit zu einer Speicherverschwendung.

5 Je kleiner die Größe der Datenzellen bzw. Datenfragmente wählt, um so größer wird der Anteil der Overhead- bzw. Header-Daten der Zellen im Vergleich zu den gespeicherten Nutzdaten. Darüber hinaus steigt die Datenbreite der in den Datenzellen enthaltenen Pointern bzw. Zeigern, die die Verket-

10 tzung der Zellen untereinander gewährleisten. Je größer die eingesetzten Speicherzellen sind, desto mehr Speicherplatz wird jedoch aufgrund der Fragmentierung verschwendet und desto größer wird die Ungenauigkeit bei der berechneten Ablaufzeit. Werden die Zellen verkleinert, desto weniger Speicher-

15 platz werden für die fragmentierten Nutzdaten jedoch desto mehr Speicherplatz wird für die Overhead- bzw. Header-Daten der Datenzellen verschwendet.

Es ist daher die Aufgabe der vorliegenden Erfindung einen Scheduler für ein Speichersystem zu schaffen, der einerseits

20 die Speicherkapazität des Speichersystems möglichst effizient ausnützt und andererseits die zur Verfügung stehenden Datenpakete mit einer sehr hohen Zeitgenauigkeit meldet.

Diese Aufgabe wird erfindungsgemäß durch einen Scheduler bzw.

25 eine Zeitablaufmeldeschaltung für ein Speichersystem mit den im Patentanspruch 1 angegebenen Merkmalen gelöst.

Die Erfindung schafft einen Scheduler für ein Speichersystem zum Zwischenspeichern von Daten, die durch mindestens eine

30 Datenverarbeitungseinheit verarbeitet werden, mit einer Schreib-Einheit zum Einschreiben von Datenobjekten in das Speichersystem,

wobei die Schreib-Einheit Datenpakete von mindestens einer Datenquelle mit einer veränderlichen Datenübertragungsrate empfängt, Attributdaten für jedes empfangene Datenpaket berechnet und die in dem Datenpaket enthaltenen Daten in das Speichersystem als eine aus miteinander verketteten Datenobjekten

35

bestehende Datenobjektkette einschreibt, die Zeigerdaten zur Verkettung der Datenobjekte, die berechneten Attributdaten und die in dem Datenpaket enthaltenen Nutzdaten umfasst, wobei die Schreib-Einheit zum Ausgleich der veränderlichen

5 Datenübertragungsrate beim Einschreiben der Datenobjekt-Kette in das Speichersystem zusätzlich Fülldaten zwischen die verketteten Datenobjekte in das Speichersystem einschreibt; einem Zähler, der durch die Schreib-Einheit beim Einschreiben der Datenobjekt-Kette in das Speichersystem entsprechend der  
10 Datenmenge der in dem Datenpaket enthaltenen Daten und der Fülldaten der Füllobjekte inkrementiert wird und mit einer Zeitablauf-Meldeeinheit, die der Datenverarbeitungseinheit meldet, dass ein in dem Speichersystem zwischengespeichertes Datenobjekt oder Füllobjekt für die Datenverarbeitungseinheit zum Auszulesen bereitstellt, wenn der Zähler einen Schwellenwert erreicht,  
15 wobei die Zeitablauf-Meldeeinheit anschließend den Zähler entsprechend den in dem bereitgestellten Objekt enthaltenen Daten dekrementiert.

20

Ein Vorteil des erfindungsgemäßen Schedulers besteht darin, dass er beliebige generische Datenpakete DP verarbeiten kann. Die ankommenden Datenpakete DP müssen kein bestimmtes Datenformat aufweisen und können variable Datenpaketgrößen besitzen. Daher kann der erfindungsgemäße Scheduler vorzugsweise  
25 auch für Ethernet-Datenpakete oder IP-Datenpakete eingesetzt werden.

30

Der erfindungsgemäße Scheduler zählt mit dem Zähler die Datenmenge der in den ankommenden Datenpaketen enthaltenen Nutzdaten und addiert dazu die Datenmenge der Fülldaten. Der Zähler zählt dabei nicht die Datenmenge der Overhead-Daten bzw. Attributdaten, die für die Datenobjekt-Kette berechnet werden.

35

Die ankommenden Datenpakete DP werden als Datenobjekt-Kette in das Speichersystem eingeschrieben. Jede Datenobjekt-Kette

besteht aus mindestens einem Datenobjekt. Die Datenobjekte der Datenobjekt-Kette sind über Zeigerdaten miteinander verkettet. Eine Datenobjekt-Kette, die mehrere Datenobjekte umfasst, besteht aus Datenobjekten unterschiedlichen Datenobjekt-Typs. Eine typische Datenobjekt-Kette besteht aus einem Ketten-Anfang-Datenobjekt (KADO), einer Vielzahl von verketteten Ketten-Mitte-Datenobjekten (KMDO) und einem Ketten-Ende-Datenobjekt (KEDO). Ist der ankommende Datenstrom so kurz, dass lediglich ein Datenobjekt benötigt wird, wird die Datenobjekt-Kette durch ein Datenobjekt gebildet, das als Ketten-Ende und Anfangs-Datenobjekt (KEADO) bezeichnet wird. Die verschiedenen Datenobjekt-Typen weisen unterschiedliche Datenformate und unterschiedliche Datengrößen auf.

Zum Ausgleich von veränderlichen Datenübertragungsraten beim Einschreiben der Datenobjekt-Kette werden in das Speichersystem durch den Scheduler zusätzlich Fülldatenobjekte zwischen die verketteten Datenobjekte eingeschrieben. Dazu sind im wesentlichen zwei verschiedene Fülldatenobjekte, nämlich ein Ein-Byte-Füll-Datenobjekt EBDFO und ein Multibyte Fülldatenobjekt MBFDO vorgesehen.

Durch die Verkettung der Datenobjekte mit variabler Datengröße wird der Speicherplatz innerhalb des Speichersystems optimal ausgenutzt. Durch das Vorsehen der Fülldatenobjekte mit variabler Größe wird zudem die Genauigkeit bei der Berechnung der Ablaufzeit optimiert, da die Fülldatenobjekte genau diejenigen Zeitspannen wiedergeben, an denen der Scheduler keine Daten von den Datenquellen empfangen hat. Die Fülldatenobjekte dienen daher zum Ausgleich der veränderlichen Datenübertragungsrate beim Einschreiben der Datenobjektkette in das Speichersystem.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers ist ein erster Datenobjekt-Typ ein Ketten-Anfangs-Datenobjekt (KADO), das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Anfangs-Datenobjekt, ein Sende-Flag, ein Zeigerda-

tenfeld zur Verkettung, ein Attributdatenfeld und ein Nutzdatenfeld aufweist.

5 Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist dieser als zweiten Datenobjekt-Typ ein Ketten-Ende-Datenobjekt (KEDO) auf, das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Ende-Datenobjekt, ein Datenfeld zur Angabe der Nutzdatenmenge und ein Nutzdatenfeld aufweist.

10 Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist dieser als dritten Datenobjekt-Typ ein Ketten-Ende- und Anfangs-Datenobjekt (KEADO) auf, das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Ende- und Anfangs-Datenobjekt, ein Sende-Flag, sein Datenfeld zur Angabe der  
15 Nutzdatenmenge, ein Attributdatenfeld und ein Nutzdatenfeld aufweist.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist dieser als vierten Datenobjekt-Typ ein Ketten-Mitte-Datenobjekt (KMDO) auf, das ein Typ-Datenfeld zur  
20 Kennzeichnung als Ketten-Mitte-Datenobjekt, ein Zeigerdatenfeld und ein Nutzdatenfeld aufweist.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist dieser als fünften Datenobjekt-Typ ein Ein-Byte-Füll-Datenobjekt (EBFDO) auf, das ein 1 Byte umfassendes  
25 Typ-Datenfeld zur Kennzeichnung als Ein-Byte-Füll-Datenobjekt aufweist.

30 Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist dieser als sechsten Datenobjekt-Typ ein Mehr-Byte-Füll-Datenobjekt (MBFDO) auf, das ein Typ-Datenfeld zur Kennzeichnung als Mehr-Byte-Füll-Datenobjekt und ein Datenfeld aufweist, welches die Fülldatenmenge angibt.

35

Bei dem erfindungsgemäßen Scheduler weist die Schreibeinheit vorzugsweise einen Steuerpfad und einen Datenpfad auf.

Der Datenpfad der Schreibeinheit weist vorzugsweise eine FIFO-Steuereinheit zum Einschreiben und Auslesen von Daten in einen daran angeschlossenen FIFO-Speicher auf.

5

Die FIFO-Steuereinheit empfängt vorzugsweise Daten von mindestens einer Datenquelle paketweise als Datenpakete.

10

Dabei weist jedes empfangene Datenpaket vorzugsweise ein Steuerdatum (SoP), welches den Anfang des Datenpakets kennzeichnet, ein Steuerdatum (EoP), welches das Ende des Datenpakets kennzeichnet und dazwischenliegende Datenpaket-Nutzdaten auf.

15

Die Datenpaket-Nutzdaten der empfangenen Datenpakete umfassen vorzugsweise jeweils Verwaltungsdaten (Header) und Informationsdaten (payload).

20

Der Steuerpfad berechnet vorzugsweise für jedes empfangene Datenpaket Attributdaten.

25

Dabei berechnet der Steuerpfad die Attributdaten vorzugsweise in Abhängigkeit von Systemeinstellungen und/oder den Verwaltungsdaten (Header) der empfangenen Datenpakete.



Der Steuerpfad speichert vorzugsweise die berechneten Attributdaten in einen Attributdatenpuffer des FIFO-Speichers.

30

Die FIFO-Steuereinheit speichert vorzugsweise die Nutzdaten eines Datenpakets in einen Nutzdatenpuffer des FIFO-Speichers.

35

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist der FIFO-Speicher für jede Datenquelle einen zugehörigen Attributdatenpuffer und einen zugehörigen Nutzdatenpuffer auf.

Die FIFO-Steuereinheit erzeugt vorzugsweise ein Fehlersignal (ERROR), wenn ein zu einer Datenquelle gehöriger Nutzdatenpuffer voll ist und keine weiteren Daten aufnimmt.

5 Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers schreibt die FIFO-Steuereinheit des Datenpfades in Abhängigkeit von Steuersignalen, die die FIFO-Steuereinheit von dem Steuerpfad der Schreibeinheit empfängt, die in dem Attributdatenpuffer zwischengespeicherten Attributdaten und  
10 die in dem Nutzdatenpuffer zwischengespeicherten Nutzdaten eines Datenpakets in das Speichersystem als eine aus miteinander verketteten Datenobjekten bestehende Datenobjekt-Kette ein.

15 Die FIFO-Steuereinheit berechnet vorzugsweise die kumulative Attributdatenmengen ( $V_{A,L}$ ) der in das Speichersystem eingeschriebenen Attributdaten eines Datenpakets.

Die FIFO-Steuereinheit berechnet vorzugsweise eine kumulative  
20 Nutzdatenmengen ( $P_L$ ) der in das Speichersystem eingeschriebenen Nutzdaten eines Datenpakets.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers enthält der Datenpfad der Schreibeinheit eine  
25 Zähleinrichtung (Timed Adress Generator), die entsprechend einer linearen Soll-Daten-Ankunfts-Kurve ( $\alpha(t)$ ) linear hochzählt

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen  
30 Schedulers weist der Datenpfad der Schreibeinheit einen effektiven Datenadressgenerator auf, der eine Zeitscheibenverteilung ( $W^*_\alpha(t)$ ) in Abhängigkeit von der berechneten kumulativen Datenmenge ( $P_L$ ) und dem von der Zähleinrichtung angelegten Zellwert berechnet.

$$W^*_{\alpha}(t) = \begin{cases} R'(t) & \text{wenn } W^*_{\alpha}(t) > \alpha(t) \\ \max[R'(t)\alpha'(t)] & \text{wenn } W^*_{\alpha}(t) = \alpha(t) \\ \alpha'(t) & \text{wenn } W^*_{\alpha}(t) < \alpha(t) \end{cases}$$

$$W^*_{\alpha}(0) = 0$$

Wobei zu allen Zeitpunkten  $\bar{t}$  gilt, sodass

5

$$\lim_{t \rightarrow \bar{t}} X(t) = \lim_{t \rightarrow \bar{t}} X(t) + h, h > 0$$

wobei die Definition des Operators „max“ folgendermaßen ausgedehnt wird:

10

$$X'(\bar{t}) = h\delta_{\bar{t}} + X''(\bar{t}), X''(\bar{t}) = \lim_{\Delta \rightarrow 0^+} \frac{X(\bar{t} + \Delta) - X(\bar{t})}{\Delta}$$

$$\max[a\delta_{\bar{t}} + A(t), B(t)] = \begin{cases} \max[A(t), B(t)] & t \neq \bar{t} \\ a\delta_{\bar{t}} + A(\bar{t}) & t = \bar{t} \end{cases}$$

$$15 \quad \max[a\delta_{\bar{t}} + A(t), b(\delta_{\bar{t}}) + B(t)] = \begin{cases} \max[A(t), B(t)] & t \neq \bar{t} \\ \max(a, b)\delta_{\bar{t}} + \max[A(\bar{t}), B(\bar{t})] & t = \bar{t} \end{cases}$$

Die Funktion  $W^*_{\alpha}(t)$  ist die kumulative Verteilung des abgegebenen Datenstroms für einen idealen Fall, bei dem ein ankommender Datenstrom mit einer kumulativen Datenverteilung  $R(t)$  bei vorgegebener Ankunftscurve bei  $\alpha(t)$  empfangen wird, wobei  $R(t)$  die Datenmenge des empfangenen Datenpakets und  $\alpha$  die lineare Soll-Daten-Ankunfts-Kurve ist.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers enthält der Datenpfad der Schreibeinheit einen Modulo-M-Addierer, der die kumulative Attributdatenmenge ( $V_{A,L}$ ) mit der berechneten Zeitschalt-Verteilung ( $W^*_{\alpha}(t)$ ) zur Erzeugung einer Datenobjekt-Adresse im Modulo-M addiert, wobei M die Speicherkapazität des Speichersystems ist.

30

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist der Datenpfad der Schreib-Steuereinheit eine aus mehreren Basis-Adress-Registern bestehendes Basis-Adress-Register-Bank auf, wobei für jede Datenquelle ein Basis-  
5 Adress-Register vorgesehen ist.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers wird bei einer Änderung der berechneten kumulativen Nutzdatenmenge ( $P_L$ ) in das Basis-Adress-Register die Anfangs-Adresse des Datenobjekts eingeschrieben.  
10

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist der Datenpfad der Schreib-Einheit eine aus mehreren Link-Adress-Registern bestehende Link-Adress-Register-Bank auf, wobei für jede Datenquelle ein Link-  
15 Adress-Register vorgesehen ist.

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers wird in das Link-Adress-Register die Adresse des zuletzt in das Speichersystem eingeschriebenen Datenobjekts für die Verkettung mit dem nächsten Datenobjekt der Datenobjekt-Kette zwischengespeichert.  
20

Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers weist der Datenpfad der Schreibeinheit einen Datenmultiplexer zum Einschreiben von Daten in das Speichersystem und einen Adressmultiplexer zum Anlegen einer Adresse an das Speichersystem auf.  
25

Dabei weist der Datenmultiplexer vorzugsweise einen ersten Eingang auf, der mit dem FIFO-Speicher zum Empfang der aus dem FIFO-Speicher ausgelesenen Attribut- und Nutzdaten und einen zweiten Eingang, der mit der Basis-Adress-Register-Bank zum Empfangen der Verkettungsdaten verbunden ist.  
30

Der Adressmultiplexer weist vorzugsweise einen ersten Eingang auf, der mit dem Ausgang des Modulo-M-Addierers zum Empfang  
35



einer Datenadresse verbunden ist, und einen zweiten Eingang, der mit der Link-Adress-Register-Bank zum Empfang einer Verkettungs-Adresse verbunden ist.

- 5 Bei einer bevorzugten Ausführungsform des erfindungsgemäßen Schedulers ist der Datenpfad der Schreib-Einheit in Abhängigkeit von einem von dem Steuerpfad generierten Steuersignals zwischen zwei Betriebsmodi umschaltbar, wobei in dem ersten Betriebsmodus zum Einschreiben von Daten-
- 10 objekten in das Speichersystem jeweils der erste Eingang des Datenmultiplexers und des Adressmultiplexers an das Speichersystem geschaltet ist, wobei in dem zweiten Betriebsmodus zum Verketten des zuletzt eingeschriebenen Datenobjekts jeweils der zweite Eingang des
- 15 Daten- und des Adressmultiplexers an das Speichersystem geschaltet ist.

Im weiteren werden bevorzugte Ausführungsformen des erfindungsgemäßen Schedulers unter Bezugnahme auf die beigefügten

20 Figuren zur Erläuterung erfindungswesentlicher Merkmale beschrieben.

Es zeigen:

- 25 Figur 1 ein Blockschaltbild einer herkömmlichen Datenverarbeitungsanordnung;

Figur 2 ein Diagramm zur Erläuterung der Funktionsweise eines Schedulers;

30

Figur 3 eine beispielhafte Anordnung zur Erläuterung eines Zeitscheiben-Scheduling-Verfahrens;

- Figur 4 einen FIFO-Speicher zur Erläuterung eines herkömmlichen Enhanced-Zeitscheiben-Verfahrens nach dem Stand der
- 35 Technik;

Figur 5 ein Blockschaltbild einer Datenverarbeitungsanordnung mit einem erfindungsgemäßen Scheduler;

Figur 6 ein Blockschaltbild der in einem erfindungsgemäßen  
5 Scheduler enthaltenen Baugruppen;

Figur 7 den Aufbau der in dem erfindungsgemäßen Scheduler enthaltenen Schreibeinheit;

10 Figur 8 ein Ablaufdiagramm zur Erläuterung des Steuervorgangs innerhalb des Steuerpfads der Schreibeinheit;

Figur 9 ein durch den erfindungsgemäßen Scheduler zu verar-  
beitenden Datenpaket;

15

Figur. 10 Datenformate der durch den erfindungsgemäßen Schedu-  
ler generierten Datenobjekte, die als Datenobjekt-Kette in  
dem Speichersystem abgespeichert werden;

20 Figur 11 ein erstes Beispiel einer in dem Speichersystem ab-  
gespeicherten Datenobjekt-Kette gemäß der Erfindung;

Figur 12 ein zweites Beispiel einer in dem Speichersystem ab-  
gespeicherten Datenobjekt-Kette gemäß der Erfindung;

25

Figur 13 ein Blockschaltbild des in dem erfindungsgemäßen  
Schedulers erhaltenen Datenpfades;

Figur 14 ein Ablaufdiagramm der FIFO-Steuereinheit zum Ein-  
30 schreiben von Nutzdaten in Nutzdatenpuffer;

Figur 15 ein Ablaufdiagramm der FIFO-Steuereinheit zur Er-  
mittlung der kumulativen Nutzdatenmenge ( $P_L$ ) und der kumula-  
tiven Attributdatenmenge;

35

Figur 16 ein Ablaufdiagramm des in der Zeitablauf-Meldeinheit ablaufenden Programms zur Dekrementierung des Zählers.

5    Figur 5 zeigt eine Datenverarbeitungs-Schaltungsanordnung, die einen erfindungsgemäßen Scheduler 2 enthält. Von mindestens einer Datenquelle 1 erhält der Scheduler 2 paketweise Daten über einen seriellen oder parallelen Datenbus 3. Der Scheduler 2 ist über Leitungen 4 mit einem Speichersystem 5  
10    verbunden. Eine Datenverarbeitungseinheit 6 liest die zu verarbeitende Daten aus dem Speichersystem 5 über einen Datenbus 7 aus. Über Leitungen 8 meldet der Scheduler 2 der Datenverarbeitungseinheit 6, dass Daten zur Verarbeitung innerhalb des Speichersystems 5 zur Verfügung stehen. Figur 5 zeigt eine  
15    Prinzipianordnung. Der Scheduler 2 kann bei alternativen Anordnungen an eine Vielzahl von Datenquellen bzw. Datenkanälen eingangsseitig angeschlossen sein. Falls mehrere Datenkanäle angeschlossen sind, werden die Daten in einem Zeit-  
20    Domain-Multiplexverfahren TDM mit jeder beliebigen Granularität geliefert. Darüber hinaus kann an das Speichersystem 5 eine Vielzahl von Datenverarbeitungseinheiten 6 angeschlossen werden, die das Speichersystem 5 als Shared Memory benutzen. Bei dem Speichersystem 5 kann es sich um ein beliebiges Speichersystem bzw. eine Speicherhierarchie mit Arbitrierung der  
25    Zugriffe handeln. Der erfindungsgemäße Scheduler 2 zeigt über Leitung 8 der Datenverarbeitungseinheit 6 an, dass Datenobjekte für die Datenverarbeitungseinheit zur weiteren Verarbeitung nach einer Ablaufzeit  $\Delta T$  bereitstehen. Der Scheduler 2 empfängt ein beliebiges Datenpaket DP von der Datenquelle.  
30    Bei dem Datenpaket kann es sich um ein generisches Datenpaket DP handeln, d.h. der erfindungsgemäße Scheduler 2 ist in der Lage, Datenpakete DP mit einem beliebigen Datenformat zu verarbeiten. Das Datenpaket DP, welches von der Datenquelle 1 stammt, weist ein Steuerdatum SoP auf, welches den Anfang des Datenpakets DP kennzeichnet, und ein Steuerdatum (EoP), welches das Ende des Datenpakets DP kennzeichnet. Das Datenpaket  
35    DP enthält zwischen den beiden Steuerdaten Nutzdaten. Diese

Nutzdaten umfassen sowohl die Datenpaket-Verwaltungsdaten bzw. Header-Daten als auch die in dem Datenpaket enthaltenen Informationsdaten bzw. die Payload. Der erfindungsgemäße Scheduler 2 ist in der Lage Datenpakete DP beliebiger Größe zu verarbeiten und der Datenverarbeitungseinheit 6 in Form von verketteten Datenobjekt-Ketten zur Verarbeitung zur Verfügung zu stellen bzw. zu melden. Bei den empfangenen Datenpaketen DP kann es sich beispielsweise um Ethernet-Datenpakete oder IP-Datenpakete handeln. Der Scheduler 2 speichert das von einer Datenquelle 1 stammende Datenpaket DP als eine Datenobjekt-Kette in das Speichersystem 5 ab. Dabei nutzt der Scheduler 2 den durch das Speichersystem 5 zur Verfügung gestellten Speicherplatz durch die Verkettung der Datenobjekte in einer Datenobjekt-Kette optimal aus. Die Zeitspanne  $\Delta T$  zwischen dem Empfang des Datenpakets und dem Zeitpunkt, an dem das letzte Datenobjekt der zugehörigen Datenobjekt-Kette der Datenverarbeitungseinheit 6 durch den Scheduler 6 gemeldet wird, wird durch eine in dem Scheduler 2 enthaltene Zeitablaufmeldeeinheit der Datenverarbeitungseinheit 6 als Ablaufzeit gemeldet. Die Ablaufzeit  $\Delta T$  ist dabei weitestgehend unempfindlich gegenüber Veränderungen bzw. Schwankungen der Datenübertragungsrate  $R'(t)$ , mit welcher die Datenpakete DP von der Datenquelle 1 an den Scheduler 2 übertragen werden. Zum Ausgleich der veränderlichen Datenübertragungsrate  $R'(t)$  schreibt der Scheduler 2 beim Einschreiben der Datenobjekt-Kette in das Speichersystem 5 zusätzlich Füllobjekte (FO) zwischen die verketteten Datenobjekte (DO) in das Speichersystem 5 ein. Die Zeitverzögerung  $\Delta T$ , mit welcher der Scheduler 2 ankommende Daten der Datenverarbeitung 6 zur Verfügung stellt, ist aufgrund der variablen Größe der zwischen den verketteten Datenobjekte (DO) gespeicherten Füllobjekte (FO) sehr genau.

Figur 6 zeigt den schaltungstechnischen Aufbau des erfindungsgemäßen Schedulers 2. Der Scheduler 2 weist bei einer bevorzugten Ausführungsform drei Baugruppen auf. Der Schedu-

ler 2 enthält eine Schreibeinheit 9, einen Zähler 10 und eine Zeitablaufmeldeeinheit 11.

Die Schreibeinheit 9 des Schedulers 2 ist über mindestens eine Datenleitung 3-0 an eine Datenquelle 1 angeschlossen. Bei einer alternativen Ausführungsform ist die Schreibeinheit 9 über eine Vielzahl von Datenleitungen an mehrere Datenquellen 1 angeschlossen. Neben der Datenleitung bzw. dem Datenbus 3-0 sind Steuerleitungen 13-3 bis 13-4 und optional die Steuerleitungen 13-1 bis 13-2 vorgesehen, mit der die Schreibeinheit 9 Steuersignale von der Datenquelle 1 empfängt. Über die Datenleitung 3-0 empfängt die Schreibeinheit 9 paketweise Daten. Die Steuerdaten, wie das Steuerdatum SoP, welches den Anfang des Datenpakets DP kennzeichnet, oder das Steuerdatum EoP, welches das Ende des Datenpakets DP kennzeichnet können entweder über die Datenleitung 3-0 mit dem Datenpaket DP übertragen werden, je nach Übertragungsprotokoll oder über separate Steuerleitungen 13-1, 13-2 wie in Figur 6 dargestellt. Über eine Steuerleitung 13-3 empfängt die Schreibeinheit das Steuerdatum, welche die Datenquelle angibt. Über eine weitere Steuerleitung 13-4 wird der Schreibeinheit 9 mitgeteilt, dass die anliegenden Daten gültig (VALID) sind. Die Datenpakete DP werden von der Datenquelle 1 mit einer Datenübertragungsrate  $\frac{dR(t)}{dt}$  übertragen und durch die Schreibeinheiten 9 empfangen.

Dabei kann die Datenübertragungsrate  $\frac{dR(t)}{dt}$  stark schwanken.

Die Datenquelle 1 sendet beispielsweise eine Vielzahl von Datenpaketen während eines Sendevorgangs und sendet dann längere Zeit keine Datenpakete DP an die Schreibeinheit 9. Die Schreibeinheit 9 empfängt die Datenpakete DP und schreibt verkettete Datenobjekte in das Speichersystem 5 ein. Die Schreibeinheit 9 empfängt die Datenpakete von mindestens einer Datenquelle 1 mit einer veränderlichen Datenübertragungsrate  $\frac{dR(t)}{dt}$ , die zusätzlich durch den Datenübertragungskanal verändert sein kann. Die Schreibeinheit 9 berechnet für jedes empfangene Datenpaket DP Attributdaten und schreibt die in

dem Datenpaket enthaltenen Nutzdaten in das Speichersystem als eine, aus miteinander verketteten Datenobjekten bestehende Datenobjekt-Kette ein. Die Datenobjekt-Kette umfasst Zeigerdaten zur Verkettung der Datenobjekte, die durch die Schreibeinheiten 9 berechneten Attributdaten und die in dem Datenpaket DP enthaltenen Nutzdaten. Zum Ausgleich der schwankenden bzw. veränderlichen Datenübertragungsrate

$\frac{dR(t)}{dt}$  schreibt die Schreibeinheit 9 beim Einschreiben der Datenobjektkette in das Speichersystem 5 zusätzlich Füllobjekte (FO) zwischen die verketteten Datenobjekte (DO) in das Speichersystem 5 ein. Die Schreibeinheit 9 ist mit dem Speichersystem 6 über eine Schreibdatenleitung 4-1, eine Schreibadressleitung 4-2, eine Write-Enable-Anforderungsleitung 4-3 und eine Write-Enable-Acknowledge-Leitung 4-4 verbunden. Über die Schreib-Enable-Request-Leitung 4-3 und die Write-Enable-Acknowledge-Leitung 4-4 übergibt die Schreibeinheit Daten an das Schreibsystem 6 in einem Hand-Shake-Verfahren. Die Schreibeinheit 9 gibt über eine Anzeigeleitung 13 im Bedarfsfalle ein Fehlersignal (ERROR) an die Datenverarbeitungseinheit 6 ab.

Der in dem Scheduler 2 enthaltene Zähler 10 wird durch die Schreibeinheit 9 beim Einschreiben der Datenobjekt-Kette in das Speichersystem 6 entsprechend der Datenmenge der in dem Datenpaket DP enthaltenen Daten und der Fülldaten der Füllobjekte über eine Leitung 14 inkrementiert.

Der erfindungsgemäße Scheduler 2 enthält neben der Schreibeinheit 9 eine Zeitablauf-Meldeeinheit 11. Die Zeitablauf-Einheit 11 meldet über Anzeigeleitungen 8-1, 8-2, 8-3 der Datenverarbeitungseinheit 6, dass ein in dem Speichersystem 5 zwischengespeichertes Datenobjekt oder Füllobjekt für die Datenverarbeitungseinheit 6 zum Auslesen bereitsteht, wenn der Zähler 10 einen Schwellenwert SW erreicht. Hierzu liest die Zeitablauf-Meldeeinheit 11 über eine Leitung 15 den Zählerstand des Zählers 10 aus und vergleicht ihn mit dem Schwellenwert. Der Schwellenwert SW ist dabei vorzugsweise über ei-

ne Programmierleitung 16 programmierbar. Wenn der Zählerstand den Schwellenwert SW erreicht, meldet die Zeitablauf-Meldeeinheit 11 der Datenverarbeitungseinheit 6 über eine Leitung 8 die Adresse des bereitstehenden Datenobjekts (DO) und über eine weitere Leitung 8-2 den Typ des bereitstehenden Datenobjekts. Optional wird die Datenverarbeitung 6 durch die Zeitablauf-Meldeeinheit 11 über weitere Leitungen 8-3 mit Informationsdaten über das gespeicherte Datenobjekt (DO) versorgt. Nachdem der Datenverarbeitungseinheit 6 durch die Zeitablauf-Meldeeinheit 11 das Bereitstehen des Datenobjekts im Speichersystem 6 gemeldet worden ist, dekrementiert die Zeitablauf-Einheit 11 anschließend dem Zähler 10 über eine Steuerleitung 16 entsprechend der Menge der in dem bereitgestellten Datenobjekt (DO) oder Füllobjekt (FO) enthaltenen Daten.

Figur 7 zeigt die in dem erfindungsgemäßen Scheduler 2 enthaltenen Schreibeinheit 9. Die Schreibeinheit 9 enthält einen Steuerpfad 9a und einen Datenpfad 9b. Der Datenpfad 9b wird durch den Steuerpfad 9a über interne Steuerleitungen 17 gesteuert.

Figur 8 zeigt das Steuerprogramm, welches innerhalb des Steuerpfades 9a abläuft. Zunächst befindet sich der Steuerpfad 9a im Ruhezustand. Er überprüft im Steuereingang 3-1, ob ein Steuerdatum SoP den Anfang eines Datenpakets DP anzeigt. Erkennt der Steuerpfad im Schritt S1, dass ein Datenpaket DP empfangen wird, geht der Steuerpfad in einen Speicherzustand über. In einem Schritt S2 überprüft der Steuerpfad 9a, ob das Steuerdatum SoP, welches den Anfang eines Datenpakets DP anzeigt, noch anliegt. Falls das Steuerdatum SoP den Anfang eines Datenpakets anzeigt wird in einem Schritt S3 geprüft, ob über die Steuerleitung 3-2 auch ein Steuerdatum EoP anliegt, welches gleichzeitig das Ende des Datenpakets DP anzeigt. In diesem Fall gibt der Steuerpfad 9a über die Steuerleitungen 17 an den Datenpfad 9b Steuersignale zum Abspeichern eines bestimmten Datenobjekts, welches als Ketten-Ende- und An-

fangs-Daten-Objekt (KEADO) bezeichnet wird, ab. Die Steuersignale zum Abspeichern der Header-Daten des Ketten-Ende- und Anfangs-Daten-Objekts KEADO werden in einem Schritt S4 an den Datenpfad 9b angelegt.

5

In einem Schritt S5 gibt der Steuerpfad 9a Steuerdaten zum Speichern der Daten als Ketten-Ende- und Anfangs-Daten-Objekt KEADO ab. Anschließend wird in einem Schritt S6 der Zähler 10 über die Leitung 14 inkrementiert.

10

Erkennt der Steuerpfad 9a im Schritt S3, dass kein Steuerdatum EoP an der Steuerleitung 3-2 anliegt, überprüft er, ob die anliegenden Daten eine bestimmte Länge L, beispielsweise 32 Byte, überschreiten. Ergibt die Längenüberprüfung im Schritt S7, dass die Datenpaketlänge mehr als 32 Byte beträgt, gibt der Steuerpfad 9a in einem Schritt S8 Steuersignale an den Datenpfad 9b über die Steuerleitungen 17 zum Speichern der Header-Daten eines zweiten Datenobjekt-Typs, nämlich eines sog. Ketten-Anfangs-Daten-Objekts (KADO) ab. In einem weiteren Schritt S9 gibt der Steuerpfad 9a Steuersignale zum Abspeichern von Daten in das Nutzdatenfeld des Ketten-Anfangs-Daten-Objekts ab. Anschließend wird der Zähler 10 entsprechend der in dem Datenobjekt abgespeicherten Datenmenge im Schritt S10 inkrementiert.

25

Erkennt der Steuerpfad im Schritt S2, dass kein Steuerdatum (SoP), welches den Anfang eines Datenpakets DP anzeigt, anliegt, wird im Schritt S11 überprüft, ob an der Steuerleitung 3-2 ein Steuerdatum (EoP) anliegt, welches das Ende des Datenpakets DP bezeichnet. Ist das Ende des Datenpakets DP erreicht, generiert der Steuerpfad 9a in einem Schritt S12 das Steuersignal für den Datenpfad 9b, welche diesen anweisen, die Header-Daten eines zweiten Datenobjekt-Typs, der als Ketten-Ende-Daten-Objekt KEADO bezeichnet wird, in dem Speichersystem 5 abzuspeichern. Anschließend generiert der Steuerpfad 9a Steuersignale für den Datenpfad 9b zum Abspeichern der Nutzdaten in das Nutzdatenfeld des Ketten-Ende-Daten-Objekts

35



in einem Schritt S13. In einem Schritt S14 wird anschließend der Zähler 10 über die Steuerleitung 14 entsprechend der Datenmenge inkrementiert, die in dem entsprechenden dem Ketten-Ende-Daten-Objekt in dem das Speichersystem 5 abgespeichert wurde.

In einem Schritt S15 wird das Ketten-Ende-Daten-Objekt (KEDO) mit dem vorangegangenen Datenobjekt verkettet. Bei dem vorangegangenen Datenobjekt (DO) handelt es sich entweder um ein Ketten-Anfangs-Daten-Objekt (KADO) oder um ein sog. Ketten-Mitte-Daten-Objekt (KMDO), die beide über Zeiger bzw. Pointer-Datenfelder verfügen, in welche die Anfangsadresse des Ketten-Ende-Daten-Objekts (KEDO) eingeschrieben wird.

Erkennt der Steuerpfad 9a im Schritt S11, dass das Steuerdatum EoP nicht anliegt, wird in einem Schritt S16 überprüft, ob die Datenlänge von beispielsweise 32 Byte überschritten worden ist. Ist dies der Fall, generiert der Steuerpfad 9a in einem Schritt S17 Steuersignale, die über die Steuerleitungen 17 an den Datenpfad 9b angelegt werden und diesen zum Speichern eines vierten Datenobjekt-Typs ansteuern. Bei dem vierten Datenobjekt-Typ handelt es sich um Ketten-Mitte-Daten-Objekt KMDO. Anschließend gibt der Steuerpfad 9a an den Datenpfad 9b in einem Schritt S18 Steuersignale zum Speichern von Daten in das Nutzdatenfeld des Ketten-Mitte-Daten-Objekts KMDO ab. In einem Schritt S19 wird der Zähler 10 entsprechen der Datenmenge der in dem Ketten-Mitte-Daten-Objekt KMDO gespeicherten Daten inkrementiert.

Ergibt die Abfrage in den Schritten S7, S16, dass die Datenlänge 32 Byte nicht erreicht worden ist, werden die Daten durch die Schreibeinheit 9 in einem Schritt S20 in einem Kanal-spezifischen Nutzdatenpuffer zwischengespeichert.

In den Schritten S1 bis S19 werden Steuerdaten zur Generierung von verschiedenen Datenobjekten DO unterschiedlichen Datenobjekt-Typs generiert. Die verschiedenen Datenobjekte,

nämlich das Ketten-Ende- und Anfangs-Daten-Objekt KEADO, das Ketten-Anfangs-Daten-Objekts KADO, das Ketten-Mitte-Daten-Objekt KMDO und das Ketten-Ende-Daten-Objekt KEDO werden durch die Schreibeinheit 9 in das Speichersystem 5 eingeschrieben und in einem Schritt S15 wird ein eingeschriebenes Datenobjekt DO mit einem Vorgänger mittels Pointer bzw. Zeigerdaten verkettet. Nachdem zwei Datenobjekte im Schritt S15 miteinander verkettet worden sind, bzw. ein Anfangs-Daten-Objekt KEADO bzw. KADO generiert worden ist, werden in den nachfolgenden Schritten die Datenlücken zwischen den beiden verketteten Datenobjekten mit Fülldaten gefüllt. Hierzu wird in einem Schritt S25 zunächst die Größe der entstandenen Datenlücke in dem Speichersystem 5 festgestellt. Besteht keine Lücke bzw. ist die Lückengröße null, geht der Vorgang über zu Schritt S22. Ist die Lückengröße genau 1 Byte groß, wird ein bestimmtes Fülldatenobjekt, nämlich ein Ein-Byte-Füll-Daten-Objekt EBFDO durch den Datenpfad 9b entsprechend den durch den Steuerpfad 9a generierten Steuerdaten in einem Schritt S22 zwischen die verketteten Datenobjekte gespeichert. Anschließend wird der Zähler 10 entsprechend der eingefügten Fülldatenmenge in einem Schritt S23 inkrementiert. Ergibt die Überprüfung in Schritt S21, dass die bestehende Datenlücke in dem Speichersystem 5 zwischen den beiden verketteten Datenobjekten DO mehr als 1 Byte beträgt, generiert der Steuerpfad 9a in einem Schritt S24 Steuerdaten zum Abspeichern eines sog. Mehr-Byte-Füll-Daten-Objekts MBFDO und gibt die Steuer-signale über die Steuerleitung 17 an den Datenpfad 9b ab. Der Datenpfad 9b speichert das Mehr-Byte-Füll-Daten-Objekt MBFDO in dem Speichersystem 5 ab. Anschließend wird in einem Schritt S25 der Zähler 10 entsprechend der gespeicherten Fülldatenmenge inkrementiert.

In einem Schritt S25 wird anschließend durch den Steuerpfad 9a überprüft, ob ein Steuerdatum EoP, welches das Ende des Datenpakets anzeigt, an die Steuerleitung 3-2 anliegt. Ist dies der Fall, kehrt der Steuerpfad 9a zunächst in den Ruhe-

zustand zurück. Ist das Ende des Datenpakets DP noch nicht erreicht, wird der Vorgang in dem Schritt S2 fortgesetzt.

Figur 10 zeigt die Datenstruktur der verschiedenen in dem Speichersystem 5 entsprechend den Steuersignalen des Steuerpfades 9a abgespeicherten Datenobjekte DO. Bei dem ersten Datenobjekt-Typ handelt es sich um ein sog. Ketten-Anfangs-Daten-Objekt KADO, das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Anfangs-Daten-Objekt (Bit-Kodierung 1 0X), ein Sende-Flag S, ein Zeigerdatenfeld zur Verkettung, ein Attributdatenfeld und ein Nutzdatenfeld für K-Byte Nutzdaten aufweist. Das Typ-Datenfeld, das Sende-Flag, das Zeigerdatenfeld sowie das Attributdatenfeld bilden die Header-Daten HD dieses Datenobjekts.

15

Das zweite Datenobjekt ist ein Ketten-Ende-Daten-Objekt KEDO, das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Ende-Daten-Objekt (Bit-Kodierung 0 1 0), ein Datenfeld zur Angabe der darin abgespeicherten Nutzdaten und ein Nutzdatenfeld aufweist. Die Länge L der in dem Ketten-Ende-Daten-Objekt KEDO abgespeicherten Nutzdaten ist variabel, wobei  $L \leq K$  ist.

20

Der dritte Datenobjekt-Typ ist ein Ketten-Ende- und Anfangs-Daten-Objekt KEADO, das ein Typ-Datenfeld zur Kennzeichnung als Ketten-Ende- und Anfangs-Daten-Objekt (Kodierung 1, 1 X), ein Datenfeld zur Angabe der darin enthaltenen Nutzdatenmenge, ein Sende-Flag S, ein Attributdatenfeld und ein Nutzdatenfeld aufweist. Die Datengröße des Ketten-Ende- und Anfangs-Daten-Objekts ist entsprechend der in dem Nutzdatenfeld gespeicherten L-Bytes variabel, wobei  $L \leq K$  ist.

25

30

Das vierte Datenobjekt ist ein sog. Ketten-Mitte-Daten-Objekt KMDO, das ein Typ-Datenfeld, nämlich die Kodierung 0, 0 zur Kennzeichnung als Ketten-Mitte-Daten-Objekt, ein Zeigerdatenfeld und ein Nutzdatenfeld aufweist. Das Nutzdatenfeld des vierten Ketten-Mitte-Daten-Objekt KMDO weist wie das Nutzda-

35

tenfeld des Ketten-Anfangs-Daten-Objekts KADO eine konstante Größe von K-Bytes Nutzdaten auf.

Die ersten vier Datenobjekte KADO, KEDO, KEADO und KMDO dienen zum Speichern der Datenpaketnutzdaten in dem Speichersystem 5. Die beiden übrigen Datenobjekte EPFO und MBFO dienen zum Auffüllen der Datenlücken zwischen den verketteten Datenobjekten DO. Das fünfte Datenobjekt ist ein Ein-Byte-Füll-Objekt EPFO, das aus einem 1 Byte umfassenden Typ-Datenfeld zur Kennzeichnung als Ein-Byte-Füll-Daten-Objekt (Bit Kodierung 0 0 1) besteht. Das sechste Datenobjekt, welches als Mehr-Byte-Füll-Objekt (Bit Kodierung 0 1 1) bezeichnet wird (MBFO) dient zum Auffüllen von größeren Datenlücken, d.h. von Datenlücken die größer sind als 1 Byte. Das Mehr-Byte-Füll-Objekt MBFO umfasst ein Typ-Datenfeld zur Kennzeichnung als Mehr-Byte-Füll-Objekt und ein Datenfeld, welches die Füll Datenmenge angibt. Als eigentliche Füll Daten können die bereits in dem Speichersystem 5 gespeicherten zufällig vorhandenen Dateien eingesetzt werden, damit es nicht nötig ist, Daten durch Schreibzyklen tatsächlich zu schreiben. Die Füll Datenmenge gibt die größte der zu füllenden Datenlücke an.

Figur 11 zeigt ein erstes Beispiel einer in dem Speichersystem 5 durch den erfindungsgemäßen Scheduler 2 abgelegten Datenobjektkette. Bei dem in Figur 11 dargestellten Beispiel besteht die Datenobjekt-Kette aus fünf Datenobjekten, nämlich einem Ketten-Anfangs-Daten-Objekt KADO, drei Ketten-Mitten-Daten-Objekt KMDO sowie einem Ketten-Ende-Daten-Objekt KEDO. Die Datenobjekte sind über Pointer bzw. Zeigerdaten miteinander verkettet. Die Zeigerdaten geben jeweils die Anfangsadresse des nächsten Datenobjekts der Datenobjektkette an. Die Datenlücken zwischen den verketteten Datenobjekten werden durch den erfindungsgemäßen Scheduler 2 durch Füll Datenobjekte d.h. durch Ein-Byte- oder Mehr-Byte-Füll-Objekte gefüllt.

Figur 12 zeigt ein weiteres Beispiel für das Abspeichern von Datenobjekt-Ketten in dem Speichersystem 5. Bei dem in Figur

12 dargestellten Beispiel sind zwei verschiedene Datenobjekt-Ketten miteinander verschachtelt in dem Speichersystem 5 abgespeichert. Eine erste Datenobjekt-Kette A besteht ebenfalls aus fünf miteinander verketteten Datenobjekten, nämlich einem Ketten-Anfangs-Daten-Objekt  $KADO_A$ , drei Ketten-Mitte-Daten-Objekten  $KMDO_A$  und einem Ketten-Ende-Daten-Objekt  $KEDO_A$ . Die zweite Datenobjekt-Kette B besteht bei dem dargestellten Beispiel aus vier Datenobjekten, nämlich einem Ketten-Anfangs-Daten-Objekt  $KADO_B$ , zwei Ketten-Mitte-Daten-Objekten  $KMDO_B$  und einem Ketten-Ende-Daten-Objekt  $KEDO_B$ . Die beiden verschiedenen Datenobjekt-Ketten A, B stammen von zwei unterschiedlichen Datenpaketen DP bzw. Datenquellen. Wie man aus Figur 12 erkennen kann, wird durch die Verschachtelung der Datenobjekt-Ketten der Speicherplatz innerhalb des Speichersystems 5 optimal ausgenutzt. Sollte aufgrund der schwankenden Datenübertragungsrate R ein Leerlauf beim Empfang der Daten DP entstehen, werden die Lücken durch die Schreibeinheit 9 mit Füllobjekten FO befüllt.

Figur 13 zeigt den schaltungstechnischen Aufbau des in Figur 7 dargestellten Datenpfades 9b. Der Datenpfad 9b enthält eine FIFO-Steuereinheit 18, die über Leitungen 19, 20 mit einem FIFO-Speicher 21 verbunden ist. Die FIFO-Steuereinheit 8 empfängt von mindestens einer Datenquelle 1 paketweise Daten über eine Datenleitung 3-0. Für jedes empfangene Datenpaket wird ein Steuerdatum SoP, welches den Anfang des Datenpakets DP kennzeichnet, und ein Steuerdatum EoP, welches das Ende des Datenpakets kennzeichnet, durch die FIFO-Steuereinheit 18 überwacht. Der Steuerpfad berechnet in Abhängigkeit von Systemeinstellungen und/oder Verwaltungsdaten des empfangenen Datenpakets DP, sog. Attributdaten. Die Attributdaten umfassen die Header- bzw. Verwaltungsdaten des empfangenen Datenpakets DP sowie weitere Information, beispielsweise über die Datenquelle, Paket-Deskriptoren und dergleichen. Der Steuerpfad speichert die berechneten Attributdaten in einen Attributdatenpuffer 22-i ab. Die in dem empfangenen Datenpaketbus DP empfangenen Nutzdaten werden durch die FIFO-Steuereinheit

18 in einen Nutzdatenpuffer 23-i des FIFO-Speichers 21 abgespeichert. Bei einer bevorzugten Ausführungsform weist der FIFO-Speicher 21 für jede Datenquelle einen Attributdatenpuffer 22-i und einen Nutzdatenpuffer 23-i auf. Erkennt die

5 Steuereinheit 18, dass ein Datenpaket DP von einer bestimmten Datenquelle 1-i stammt, werden die berechneten Attributdaten in den zugehörigen Attributdatenpuffer 22-i und die Nutzdaten in den zugehörigen Nutzdatenpuffer 23-i eingeschrieben. Die

10 FIFO-Steuereinheit 18 erzeugt ein Fehlersignal (ERROR), wenn ein zu einer Datenquelle 1 gehöriger Nutzdatenpuffer 23-i voll ist und weitere Datenpakete bzw. Daten von dieser Datenquelle empfangen werden. Das Fehlersignal (ERROR) wird über

15 eine Steuerleitung 13 abgegeben. Die FIFO-Steuereinheit 18 empfängt von dem Steuerpfad 9a über Steuersignalleitungen 17-1, 17-2, 17-3 Steuersignale. Über eine Steuerleitung 17-1 empfängt die FIFO-Steuereinheit 18 die in den Schritten S4, S8, S12, S17 generierten Steuersignale zum Einschreiben der

20 Header-Daten HD der vier verschiedenen Datenobjekte KEADO, KADO, KEDO, KMDO. Über eine Steuerleitung 17-2 erhält die FIFO-Steuereinheit 18 die Header-Source. Durch eine Steuerleitung 17-3 empfängt die FIFO-Steuereinheit 18 schließlich ein Store-Data-Steuersignal zur Selektion von Attribut- oder

25 Content- bzw. Nutzdaten. Der FIFO-Speicher 21 ist über Datenleitungen 24 mit einem ersten Eingang 25 eines Datenmultiplexers 26 verbunden. Über die Datenleitungen 24 werden zur Generierung von Datenobjekten DO die Attributdatenpuffer 22 und Nutzdatenpuffer 23 des FIFO-Speichers 1 entsprechend den

30 Steuerbefehlen in das Speichersystem 5 ausgelesen. Die FIFO-Steuereinheit 18 berechnet die kumulativen Attributdatenmengen  $V_{A,C}$  der in das Speichersystem 5 eingeschriebenen Attributdaten eines Datenpakets DP. Die kumulative Attributdatenmenge  $V_{A,L}$  wird dabei wie folgt ermittelt.

Bei der Paketierung wird die kumulative Datenpaketlänge  $L_n$

35 wie folgt definiert:

$$\begin{cases} L_0 = 0 \\ L_n \geq L_{n-1}, \forall n > 0 \end{cases}$$

wobei

$$5 \quad \sup_{n>0} \{L_{n+1} - L_n\} \in \mathbb{R}$$

Der Wert  $L_n - L_{n-1}$  wird als Länge des n-ten Datenpakets interpretiert.

- 10 Die kumulative Datenpaketlänge  $P_L$  ergibt sich als Funktion der kumulativen Paketlänge  $L_n$  wie folgt:

$$P_L(x) = \sup_{\substack{n \geq 0 \\ L_n \leq x}} \{L_n\}$$

- 15 Die Funktion  $P_L$  wird als die lokale Transferfunktion angesehen.

Bei der Paketierung wird ein Datenstrom mit einer kumulativen Verteilung  $R(t)$  empfangen und ein Datenstrom mit einer kumulativen Verteilung  $P_L(R(t))$  erzeugt.

20

Um die berechneten Attributdaten, wie beispielsweise Datenobjekt-Zeiger oder Zeitstempel, zu berücksichtigen wird eine weitere kumulative Attributlänge  $A_n$ , die analog zur Paketlänge ist, bestimmt, welche die Attributdaten berücksichtigt. Die Datenmenge  $A_n - A_{n-1}$  bildet die Datengröße für Attributdaten des n-ten Datenpakets.

25

Die kumulative Attributdatenmenge  $V_{A,L}$  ergibt sich aus den kumulativen Paketlängen  $A_n$  und  $L_n$  gemäß folgender Gleichung:

30

$$V_{A,L}(x) = \sup_{\substack{n \geq 0 \\ L_n \leq x}} \{A_n\}$$

Die Funktion  $V_{A,L}$  bildet in diesem Falle als die ideale Transferfunktion.

Die Berechnungseinheit zur Berechnung der Attributdaten ergibt zwei kumulative Datenpaketlängen  $A_n$  und  $L_n$  und einen charakteristischen Wert  $V_{A,L}$ . Der entsprechend obiger Gleichung berechnete kumulative Attributdatenmengenwert  $V_{A,L}$  wird von der Steuereinheit 18 über eine Leitung 27 und eine Verzögerungsschaltung 28, die zur Zeitsynchronisierung dient, einem ersten Eingang 29 eines Modulo-M-Addierers 30 zugeführt. Die berechnete kumulative Nutzdatenmenge  $P_L$  wird von der Steuereinheit 18 über eine Leitung 31 einem ersten Eingang 32 eines effektiven Datenadressgenerators (EDAC) 33 zugeführt. Der effektive Datenadressgenerator 33 berechnet eine Zeitscheiben- bzw. Timing-Wheel-Verteilung bzw. Distribution. Die Zeitscheiben-Distribution stellt die kumulative Verteilung des abgehenden Datenflusses da. Die Zeitscheiben-Verteilung ( $W^*_\alpha(t)$ ) wird in Abhängigkeit von der durch die Steuereinheit 18 berechneten kumulativen Datenmenge ( $P_L$ ) und der von einer Zähleinrichtung 34 abgegebenen Datenankunftskurve  $\alpha$  berechnet. Hierzu weist der Datenpfad 9a einen Hochzähl- bzw. eine Zähleinrichtung 34 auf, die entsprechend einer idealen Soll-Datenankunftskurve einen ansteigenden Zählwert über eine Leitung 35 an einen zweiten Eingang 36 des effektiven Datenadressgenerators 33 liefert. Die Zeitscheiben-Verteilung wird entsprechend folgender Gleichung berechnet.

$$W^*_\alpha(t) = \begin{cases} R'(t) & \text{wenn } W^*_\alpha(t) > \alpha(t) \\ \max[R'(t), \alpha'(t)] & \text{wenn } W^*_\alpha(t) = \alpha(t) \\ \alpha'(t) & \text{wenn } W^*_\alpha(t) < \alpha(t) \end{cases}$$

$$W^*_\alpha(0) = 0$$

$$W^*_\alpha(0) = 0$$

Wobei zu allen Zeitpunkten  $\bar{t}$  gilt, sodass

$$\lim_{t \rightarrow \bar{t}} X(t) = \lim_{t \rightarrow \bar{t}} X(t) + h, h > 0$$



wobei die Definition des Operators „max“ folgendermaßen ausgedehnt wird:

$$X'(\bar{t}) = h\delta_i + X''(\bar{t}), X''(\bar{t}) = \lim_{\Delta \rightarrow 0^+} \frac{X(\bar{t} + \Delta) - X(\bar{t})}{\Delta}$$

5

$$\max[a\delta_i + A(t), B(t)] = \begin{cases} \max[A(t), B(t)] & t \neq \bar{t} \\ a\delta_i + A(\bar{t}) & t = \bar{t} \end{cases}$$

$$\max[a\delta_i + A(t), b(\delta_i) + B(t)] = \begin{cases} \max[A(t), B(t)] & t \neq \bar{t} \\ \max(a, b)\delta_i + \max[A(\bar{t}), B(\bar{t})] & t = \bar{t} \end{cases}$$

Die Funktion  $W^*\alpha(t)$  ist die kumulative Verteilung des abgegebenen Datenstroms für einen idealen Fall, bei dem ein ankommender Datenstrom mit einer kumulativen Datenverteilung  $R(t)$  bei vorgegebener Ankunftscurve bei  $\alpha(t)$  empfangen wird, wobei  $R(t)$  die Datenmenge des empfangenen Datenpakets und  $\alpha$  die lineare Soll-Daten-Ankunfts-Kurve ist.

Die Zeitscheiben-Verteilung ( $W^*\alpha(t)$ ) bildet die kumulative Verteilung des ausgehenden Datenstroms für eine ideale Vorrichtung, die einen ankommenden Datenstrom mit einer kumulativen Verteilung  $R(t)$  zum Erkennen der Datenankunftscurve  $\alpha$  empfängt. Die Zeitscheibe stellt dabei einen Idealfall dar, der bei einer vorgegebenen Datenankunftscurve  $\alpha$  und einer kumulativen Datenverteilung des empfangenen Datenstroms  $R(t)$  einen abgehenden Datenstrom ( $W^*\alpha(t)$ ) erzeugt. Die durch den effektiven Adressgenerator 33 erzeugte Zeitscheiben-Verteilung ( $W^*\alpha(t)$ ) wird über eine Leitung 37 an einen zweiten Eingang 38 des Modulo-M-Addierers 30 angelegt. Der Modulo-M-Addierer 38 addiert die kumulative Attributdatenmenge  $V_{A,L}$  mit der berechneten Zeitscheiben-Verteilung ( $W^*\alpha(t)$ ), die zur Erzeugung einer Datenobjektadresse, die über Leitungen 39 an einen ersten Eingang 40 eines Adressmultiplexers 41 angelegt wird. Bei der Modulo-M-Addition stellt  $M$  die Speicherkapazität des Speichersystems 5 dar. Ist die Summe der beiden

an den Eingängen 29, 38 anliegenden Werte größer als M, erzeugt der Modulo-M-Addierer ein Überlauf-Anzeigesignal WRAP und gibt dies über eine Steuerleitung 42 ab.

- 5 Das Überlauf-Anzeigesignal führt dazu, dass die FIFO-  
Steuereinheit 18, die Zähleinrichtung 34 und der effektive  
Adressengenerator 33 ihren jeweiligen Datenausgangswert ( $V_{A,L}$ ,  
 $P_L, \alpha, W^*_2$ ) um M vermindern.
- 10 Der Datenpfad 9a der Schreibsteuereinheit 9 weist einen aus  
mehreren Basis-Adress-Registern BAR bestehende Basis-Adress-  
Register-Bank 43 auf. Dabei ist für jede Datenquelle 1 vor-  
zugsweise ein Basis-Adress-Register BAR vorgesehen. Die Ba-  
sis-Adress-Register-Bank 43 erhält über eine Datenleitung 44  
15 das Ausgangssignal des Modul-M-Addierers 30.

- Über eine Leitung 45 wird der Basis-Adress-Register-Bank 43  
ferner ein Steuersignal von der FIFO-Steuereinheit 18 zuge-  
führt, wenn sich der Wert der berechneten kumulativen Nutzda-  
20 tenmenge  $P_L$  ändert. Dieses Signal dient als Write-Enable WE  
für die Basis-Adress-Register-Bank und zeigt an, dass durch  
die FIFO-Steuereinheit 18 ein neues Datenobjekt in das Spei-  
chersystem 5 abgespeichert wird. Über eine Leitung 46 und ei-  
ne Verzögerungsschaltung 47 zur Zeitsynchronisierung empfängt  
25 die Basis-Adress-Register-Bank 43 die Quell- bzw. Schreibad-  
resse von der FIFO-Steuereinheit 18. Jedes Mal, wenn ein Da-  
tenobjekt DO generiert wird, wird das Basis-Adress-Register  
der zugehörigen Datenquelle 1 auf die Anfangsadresse des Da-  
tenobjekts DO gesetzt. Die in dem Basis-Adress-Register BAR  
30 zwischengespeicherte Basis-Adresse wird über Leitungen 48 an  
einen zweiten Eingang 49 des Datenmultiplexers 26 angelegt.

- Neben der Basis-Adress-Register-Bank 43 enthält der Datenpfad  
9 eine Link-Adress-Register-Bank 50, die zur Verkettung der  
35 eingeschriebenen Datenobjekte DO vorgesehen ist. Vorzugsweise  
ist für jede Datenquelle 1 ein Link-Adress-Register LAR in-  
nerhalb der Link-Adress-Register-Bank 50 vorgesehen. In dem

Link-Adress-Register wird die Adresse des zuletzt in das Speichersystem 5 eingeschriebenen Datenobjekts DO für die Verkettung mit dem nächsten Datenobjekt DO der Datenobjektkette DOK zwischengespeichert. Die zwischengespeicherte Adresse wird über Datenleitungen 51 an einen zweiten Eingang 52 des Adressmultiplexers 41 angelegt. Die Link-Adress-Register-Bank 50 empfängt über eine Steuerleitung 17-6 ein Enable-Signal von dem Steuerpfad 9a. Über eine Steuerleitung 17-5 wird eine Link-Adress-Quelle der Basis-Adress-Register-Bank 3 und der Link-Adress-Register-Bank 50 zugeführt. Der Datenmultiplexer 26 weist einen Steuereingang 53 zum Empfang eines Link-Adress-Selektor-Signals über eine Steuerleitung 17-4 auf. Der Adressmultiplexer 41 weist ebenfalls einen Steuereingang 54 auf zum Empfang des Link-Adress-Selektor-Steuersignals.

Der Datenpfad 9a der Schreibeinheit 9 ist in Abhängigkeit von dem Steuersignal zwischen zwei Betriebsmodi umschaltbar, wobei in dem ersten Betriebsmodus die Datenobjekte (DO) in das Speichersystem 5 eingeschrieben werden und in dem zweiten Betriebsmodus die eingeschriebenen Datenobjekte (DO) miteinander verkettet werden. In dem ersten Betriebsmodus werden die an dem ersten Eingang 25 des Datenmultiplexers 26 anliegenden Nutz- und Attributdaten und die über den ersten Dateneingang 40 des Adressmultiplexers 41 durch den Modulo-M-Addierer 30 berechnete Adresse durchgeschaltet und eingeschrieben. Nachdem ein Datenobjekt  $DO_n$  eingeschrieben worden ist, wird es mit dem vorherigen Datenobjekt  $DO_{n-1}$  verkettet, indem beide Multiplexer 26, 41 auf den zweiten Eingang 49 bzw. 52 mittels des Steuersignals Link-Adress-Selektor umgeschaltet werden. Nach dem Umschalten wird das in dem Basis-Adress-Register BAR gespeicherte Datum an die in dem Link-Adress-Register LAR gespeicherte Adresse zur Verkettung der beiden Datenobjekte  $DO_n$ ,  $DO_{n-1}$  geschrieben.

Figur 14 zeigt ein Ablaufdiagramm zum Einschreiben der Nutzdaten in die Nutzdatenpuffer 23-i des FIFO-Speichers 21. Die

FIFO-Steuereinheit 8 befindet sich zunächst in einem Ruhezustand. Sobald die FIFO-Steuereinheit 18 über die Steuerleitung 3-4 ein Gültig- bzw. VALID-Signal in dem Schritt S1 erhält, prüft die Steuereinheit 18 im Schritt S2, ob der zu der Datenquelle 1 zugehörige Nutzdatenpuffer 23-i bereits gefüllt ist. Ist dies der Fall, gibt die FIFO-Steuereinheit 18 in einem Schritt S3 ein Fehlersignal über die Anzeigeleitung 13 ab. Ist der Nutzdatenpuffer 23-i noch nicht gefüllt, werden die Nutzdaten des empfangenen Datenpakets DP in den für die Datenquelle 1-x vorgesehenen Nutzdatenpuffer 23-i in einem Schritt S4 eingeschrieben. Anschließend kehrt die FIFO-Steuereinheit 18 in den Ruhezustand zurück.

Figur 15 zeigt das Auslesen von Daten aus dem FIFO-Speicher 21 durch die FIFO-Steuereinheit 18 zum Abspeichern in dem Speichersystem 5 über den Datenmultiplexer 26. Die FIFO-Steuereinheit 18 befindet sich zunächst in einem Ruhezustand. Erhält die FIFO-Steuereinheit 18 über eine Steuerleitung 17-3 ein POP-Steuersignal von dem Steuerpfad 9a, prüft die Steuereinheit 18 in einem Schritt S2, ob Attribut- oder Nutzdaten auszulesen sind. Handelt es sich bei den auszulesenden Daten um Attributdaten werden in einem Schritt S3 Attributdaten aus dem Attributdatenpuffer 22-i entsprechend der über die Steuerleitung 17-3 angegebenen Datenlänge L ausgelesen. Anschließend wird durch die Steuereinheit 18 die kumulative Attributdatenmenge  $V_{A,L}$  in einem Schritt S4 inkrementiert. Der inkrementierte Wert wird beispielsweise in einem Register innerhalb der Steuereinheit 18 zwischengespeichert. Ergibt die Abfrage in Schritt S2, dass die auszulesenden Daten Nutzdaten sind, werden in einem Schritt S5 die Nutzdaten entsprechend der angegebenen Datenlänge L aus dem zu der Datenquelle gehörigen Nutzdatenpuffer 23-i ausgelesen. Anschließend wird in einem Schritt S6 die kumulative Nutzdatenmenge  $P_L$  durch die FIFO-Steuereinheit 18 inkrementiert und vorzugsweise in einem Register zwischengespeichert.

Figur 16 zeigt den in der Zeitablauf-Meldeeinheit 11 des Schedulers 2 ablaufenden Vorgang. Zunächst befindet sich die Zeitablauf-Meldeeinheit 11 in einem Ruhezustand. Die Zeitablauf-Meldeeinheit 11 fragt über eine Leitung 15 den Zähler-

5 stand des Zählers 10 in einem Schritt S1 ab und vergleicht ihn mit einem vorzugsweise programmierbaren Schwellenwert SW im Schritt S1. Überschreitet der Zählerwert den Schwellenwert SW, wird in einem Schritt S2 der Datenobjekt-Header (HD) aus dem Speichersystem 6 über eine Leitung 4-8 bei der Anfangsadresse des letzten Datenobjekts der Datenobjektkette ausgelesen.

10 In dem Schritt S3 wird die Datenobjektgröße des Datenobjekts bestimmt. In einem nächsten Schritt S4 wird die Datenmenge der in dem Datenobjekt enthaltenen Nutzdaten ermittelt. In einem weiteren Schritt S5 wird die „Last Address“ um die

15 Datenmenge „Objekt-Data-Size“ inkrementiert. Anschließend wird in einem Schritt S6 der Zähler 10 entsprechend der Datenmenge der in dem letzten Datenobjekt DO enthaltenen Nutzdaten dekrementiert.

## Patentansprüche

1. Scheduler für ein Speichersystem (5) zum Zwischenspeichern von Daten, die durch mindestens eine Datenverarbeitungseinheit (6) verarbeitet werden, mit:

(a) einer Schreib-Einheit (9) zum Einschreiben von Datenobjekten in das Speichersystem (5), wobei die Schreib-Einheit Datenpakete (DP) von mindestens einer Datenquelle (1) mit einer veränderlichen Datenübertragungsrate empfängt, Attributdaten für jedes empfangene Datenpaket (DP) berechnet und die in dem Datenpaket (DP) enthaltenen Daten in das Speichersystem (5) als eine aus miteinander verketteten Datenobjekten (DO) bestehende Datenobjektkette einschreibt, die Zeigerdaten (Pointer) zur Verkettung der Datenobjekte (DO), die berechneten Attributdaten und die in dem Datenpaket (DP) enthaltenen Nutzdaten (ND) umfasst,

wobei die Schreib-Einheit (9) zum Ausgleich der veränderlichen Datenübertragungsrate beim Einschreiben der Datenobjektkette in das Speichersystem (5) zusätzlich Füllobjekte (FO) zwischen die verketteten Datenobjekte (DO) in das Speichersystem (5) einschreibt;

(b) einem Zähler (10), der durch die Schreib-Einheit (9) beim Einschreiben der Datenobjekt-Kette in das Speichersystem (5) entsprechend der Datenmenge der in dem Datenpaket (DP) enthaltenen Daten und der Fülldaten der Füllobjekte inkrementiert wird und mit

(c) einer Zeitablauf-Meldeeinheit (11), die der Datenverarbeitungseinheit (6) meldet, dass ein in dem Speichersystem (5) zwischengespeichertes Datenobjekt (DO) oder Füllobjekt (FO) für die Datenverarbeitungseinheit zum Auslesen bereitsteht, wenn der Zähler (10) einen Schwellenwert (SW) erreicht,

wobei die Zeitablauf-Meldeeinheit (11) anschließend den Zähler (10) entsprechend den in dem bereitgestellten Objekt enthaltenen Daten dekrementiert.

2. Scheduler nach Anspruch 1,  
dadurch gekennzeichnet,  
dass die Datenobjekt-Kette aus verketteten Datenobjekten be-  
steht, die unterschiedliche Datenobjekt-Typen aufweisen.

3. Scheduler nach Anspruch 2,  
dadurch gekennzeichnet,  
dass ein erster Datenobjekt-Typ ein Ketten-Anfangs-  
Datenobjekt (KADO) ist, dass ein Typdatenfeld zur Kennzeich-  
nung als Ketten-Anfangs-Datenobjekt,  
ein Sende-Flag,  
ein Zeigerdatenfeld zur Verkettung,  
ein Attributdatenfeld und ein Nutzdatenfeld aufweist.

4. Scheduler nach Anspruch 2,  
dadurch gekennzeichnet,  
dass ein zweiter Datenobjekt-Typ ein Ketten-Ende-Datenobjekt  
(KEDO) ist, das ein Typdatenfeld zur Kennzeichnung als Ket-  
ten-Ende-Datenobjekt ein Datenfeld zur Angabe der Nutzdaten-  
menge und ein Nutzdatenfeld aufweist.

5. Scheduler nach Anspruch 2,  
dadurch gekennzeichnet,  
dass ein dritter Datenobjekt-Typ ein Ketten-Ende- und Anfang-  
Datenobjekt (KEADO), das ein Typdatenfeld zur Kennzeichnung  
als Ketten-Ende- und Anfangs-Datenobjekt ein Datenfeld zur  
Angabe der Nutzdatenmenge, ein Sende-Flag, ein Attributda-  
tenfeld und ein Nutzdatenfeld aufweist.

6. Scheduler nach Anspruch 2,  
dadurch gekennzeichnet,  
dass ein vierter Datenobjekt-Typ ein Ketten-Mitte-Datenobjekt  
(KMDO) ist, das ein Typdatenfeld zur Kennzeichnung als Ket-  
ten-Mitte-Datenobjekt, ein Zeigerdatenfeld und ein Nutzdaten-  
feld aufweist.

7. Scheduler nach Anspruch 2,

d a d u r c h g e k e n n z e i c h n e t ,

dass ein fünfter Datenobjekt-Typ ein Ein-Byte-Füll-Objekt (EBFO) ist, das 1 Byte umfassender Typ-Datenfeld zur Kenn-

5 zeichnung als EIN-BYTE-Füll-Objekt (EBFO) aufweist.

8. Scheduler nach Anspruch 2,

d a d u r c h g e k e n n z e i c h n e t ,

dass sechster Datenobjekt-Typ ein Mehr-Byte-Füll-Objekt

10 (MBFO) ist, das ein Typ-Datenfeld zur Kennzeichnung als Mehr-Byte Füllobjekt (FO) und ein Datenfeld, welches die Fülldatenmenge angibt, aufweist.

9. Scheduler nach Anspruch 1,

15 d a d u r c h g e k e n n z e i c h n e t ,

dass die Schreib-Einheit (9) eine Steuerpfad (9a) und einen Datenpfad (9b) enthält.

10. Scheduler nach Anspruch 9,

20 d a d u r c h g e k e n n z e i c h n e t ,

dass der Datenpfad (9b) der Schreib-Einheit (9) eine FIFO-Steuereinheit (18) zum Einschreiben und Auslesen von Daten in einem daran angeschlossenen FIFO-Speicher (21) enthält.

25 11. Scheduler nach Anspruch 10,

d d a d u r c h g e k e n n z e i c h n e t ,

dass die FIFO-Steuereinheit (18) Daten von mindestens einer Datenquelle (1) paketweise als Datenpakete (DP) empfängt.

30 12. Scheduler nach Anspruch 11,

d a d u r c h g e k e n n z e i c h n e t ,

dass für jedes empfangene Datenpaket (DP) ein Steuerdatum (SoP), welches den Anfang des Datenpaketes (DP) kennzeichnet und ein Steuerdatum (EoP), welches das Ende des Datenpakets

35 (DP) kennzeichnet, vorgesehen ist.



13. Scheduler nach Anspruch 12,  
dadurch gekennzeichnet,  
dass die Datenpaket-Nutzdaten (ND) des empfangenen Datenpake-  
te (DP) jeweils Verwaltungsdaten (Header) und Informationsda-  
ten (payload) umfassen.
14. Scheduler nach Anspruch 10,  
dadurch gekennzeichnet,  
dass die FIFO-Steuereinheit (18) für jedes empfangene Daten-  
paket (DP) Attributdaten berechnet.
15. Scheduler nach Anspruch 14,  
dadurch gekennzeichnet,  
dass der Steuerpfad die Attributdaten in Abhängigkeit von  
Systemeinstellungen und den Verwaltungsdaten (Header) der Da-  
tenpakete (DP) berechnet.
16. Scheduler nach Anspruch 14,  
dadurch gekennzeichnet,  
dass die FIFO-Steuereinheit (18) die berechneten Attributda-  
ten in einen Attributdatenpuffer (22) des FIFO-Speichers (21)  
zwischenspeichert.
17. Scheduler nach Anspruch 10,  
dadurch gekennzeichnet,  
dass die FIFO-Steuereinheit (18) die Nutzdaten eines Datenpa-  
kets (DP) in einen Nutzdatenpuffer (23) des FIFO-Speichers  
(21) zwischenspeichert.
18. Scheduler nach einem der vorangehenden Ansprüche,  
dadurch gekennzeichnet,  
dass der FIFO-Speicher (21) für jede Datenquelle (1) einen  
Attributdatenpuffer (22) und einen Nutzdatenpuffer (23) auf-  
weist.
19. Scheduler nach Anspruch 18,  
dadurch gekennzeichnet,

dass die FIFO-Steuereinheit (18) ein Fehlersignal (ERROR) erzeugt, wenn ein zu einer Datenquelle gehöriger Nutzdatenpuffer (23) voll ist und keine weiteren Daten aufnimmt.

5 20. Scheduler nach einem der vorangehenden Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t ,  
dass die FIFO-Steuereinheit (18) des Datenpfades (9b) in Ab-  
hängigkeit von Steuersignalen, die die FIFO-Steuereinheit  
(18) von dem Steuerpfad (9a) der Schreibeinheit (9) empfängt,  
10 die in dem (22) zwischengespeicherten Attributdaten und die  
in dem Nutzdatenpuffer (23) zwischengespeicherten Nutzdaten  
eines Datenpakets (DP) in das Speichersystem (5) als eine aus  
miteinander verketteten Datenobjekten (DO) bestehende Daten-  
objekt-Kette einschreibt.

15

21. Scheduler nach Anspruch 10,  
d a d u r c h g e k e n n z e i c h n e t ,  
dass die FIFO-Steuereinheit (18) eine kumulative Attribut-  
Datenmenge ( $V_{A,c}$ ) der in das Speichersystem (5) eingeschriebe-  
20 nen Attributdaten eines Datenpakets (DP) erfasst.

22. Scheduler nach Anspruch 10,  
d a d u r c h g e k e n n z e i c h n e t ,  
dass die FIFO-Steuereinheit (18) die kumulative Nutz-  
25 Datenmenge ( $P_L$ ) der in das Speichersystem (5) eingeschriebe-  
nen Nutzdaten eines Datenpakets (DP) erfasst.

23. Scheduler nach Anspruch 9,  
d a d u r c h g e k e n n z e i c h n e t ,  
30 dass der Datenpfad (9b) der Schreibeinheit (9) eine Zählein-  
richtung (34) (Timed Address Generator) enthält, die entspre-  
chend einer linearen Soll-Datenankunftskurve ( $\alpha(t)$ ) linear  
hochzählt.

35 24. Scheduler nach Anspruch 9,  
d a d u r c h g e k e n n z e i c h n e t ,

dass der Datenpfad (9b) der Schreibeinheit (9) einen effektiven Datenadressgenerator (33) aufweist, der eine Zeitscheibenverteilung ( $W^*_\alpha(t)$ ) in Abhängigkeit von der berechneten kumulativen Datenmenge ( $P_L$ ) und dem von der Zähleinrichtung (34) angelegten Zählwert folgendermaßen berechnet:

$$W^*_\alpha(t) = \begin{cases} R'(t) & \text{wenn } W^*_\alpha(t) > \alpha(t) \\ \max[R'(t)\alpha'(t)] & \text{wenn } W^*_\alpha(t) = \alpha(t) \\ \alpha'(t) & \text{wenn } W^*_\alpha(t) < \alpha(t) \end{cases}$$

wobei  $R(t)$  die Datenmenge des empfangenen Datenpakets (DP) und  $\alpha$  die lineare Soll-Datenankunftskurve ist.

25. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass der Datenpfad (9b) der Schreib-Einheit (9) einen Modulo M-Addierer (30) enthält, der die kumulative Attribut-Datenmenge ( $V_{A,L}$ ) mit der berechneten Zeitscheibenverteilung ( $W^*_\alpha(t)$ ) zur Erzeugung einer Datenobjektadresse Modulo M addiert, wobei M die Speicherkapazität des Speichersystems (5) ist.

26. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass der Datenpfad (9b) der Schreib-Steuereinheit (9) eine aus mehreren Basis-Adress-Registern bestehende Basis-Adress-Register-Bank (43) aufweist, wobei für jede Datenquelle (1) ein Basis-Adress-Register (BAR) vorgesehen ist.

27. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass bei einer Änderung der berechneten kumulativen Nutzdatenmenge ( $P_L$ ) in das Basis-Adress-Register (BAR) die Anfangsadresse des Datenobjekts (DO) eingeschrieben wird.

28. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet,

dass der Datenpfad (9b) der Schreibeinheit (9) eine aus mehreren Link-Adress-Registern bestehende Link-Adress-Register-Bank (43) aufweist, wobei für jede Datenquelle (1) ein Link-Adress-Register (LAR) vorgesehen ist.

5

29. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass in dem Link-Adress-Register (LAR) die Adresse des zuletzt in das Speichersystem (5) eingeschriebenen Datenobjekts (DO) für die Verkettung mit dem nächsten Datenobjekt (DO) der Datenobjekt-Kette zwischengespeichert ist.

10

30. Scheduler nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass der Datenpfad (9b) der Schreibeinheit (9) einen Datenmultiplexer (26) zum Einschreiben von Daten in das Speichersystem (5) und einen Adress-Multiplexer (41) zum Anlegen einer Adresse an das Speichersystem (5) aufweist.

20

31. Scheduler nach Anspruch 30, dadurch gekennzeichnet, dass der Daten-Multiplexer (26) einen ersten Eingang (25), der mit dem FIFO-Speicher (21) zum Empfang der aus dem FIFO- (21) ausgelesenen Attribut- und Nutzdaten und einen zweiten Eingang (49) aufweist, der mit der Basis-Adress-Register-Bank (43) zum Empfang der Verkettungsdaten verbunden ist.

25

32. Scheduler nach Anspruch 30, dadurch gekennzeichnet, dass der Adress-Multiplexer (41) einen ersten Eingang (40), der mit dem Ausgang des Modulo-M-Addierers (30) zum Empfang einer Datenadresse verbunden ist, und einen zweiten Eingang (52) aufweist, der mit der Link-Adress-Register-Bank (43) zum Empfang einer Verkettungsadresse verbunden ist.

30

35

33. Scheduler nach einem der vorangehenden Ansprüche,  
dadurch gekennzeichnet,  
dass der Datenpfad (9b) der Schreibeinheit (9) in Abhängig-  
keit von einem von dem Steuerpfad (9a) generierten Steuersig-  
5 nal zwischen zwei Betriebsmodi umschaltbar ist, wobei in dem  
ersten Betriebsmodus zum Einschreiben von Datenobjekten in  
das Speichersystem (5) jeweils der erste Eingang (25, 40) des  
Daten-Multiplexers (26) und des Adress-Multiplexers (41) an  
das Speichersystem (5) geschaltet ist,  
10 wobei in dem zweiten Betriebsmodus zum Verketteten des zuletzt  
eingeschriebenen Datenobjekts (DO) jeweils der zweite Eingang  
(49, 52) des Daten- und des Adress-Multiplexers (26, 41) an  
das Speichersystem (5) geschaltet ist.

## Zusammenfassung

Scheduler (2) für ein Speichersystem (5) zum Zwischenspeichern von Daten, die durch mindestens eine Datenverarbeitungseinheit (6) verarbeitet werden, mit einer Schreib-Einheit (9) zum Einschreiben von Datenobjekten in das Speichersystem (5), wobei die Schreib-Einheit (9) Datenpakete (DP) von mindestens einer Datenquelle (1) mit einer veränderlichen Datenübertragungsrate empfängt, Attributdaten für jedes empfangene Datenpaket (DP) berechnet und die in dem Datenpaket (DP) enthaltenen Daten in das Speichersystem (5) als eine aus miteinander verketteten Datenobjekten (DO) bestehende Datenobjektkette einschreibt, die Zeigerdaten zur Verkettung der Datenobjekte (DO), die berechneten Attributdaten und die in dem Datenpaket (DP) enthaltenen Nutzdaten umfasst, wobei die Schreib-Einheit (9) zum Ausgleich der veränderlichen Datenübertragungsrate beim Einschreiben der Datenobjekt-Kette in das Speichersystem (5) zusätzlich Füllobjekte (FO) zwischen die verketteten Datenobjekte (DO) in das Speichersystem (5) einschreibt; einem Zähler (10), der durch die Schreib-Einheit (9) beim Einschreiben der Datenobjekt-Kette in das Speichersystem (5) entsprechend der Datenmenge der in dem Datenpaket (DP) enthaltenen Daten und der Fülldaten der Füllobjekte (FO) inkrementiert wird und mit einer Zeitablauf-Meldeeinheit (11), die der Datenverarbeitungseinheit (6) meldet, dass ein in dem Speichersystem (5) zwischengespeichertes Datenobjekt (DO) oder Füllobjekt (FO) für die Datenverarbeitungseinheit (6) zum Auslesen bereitstellt, wenn der Zähler (10) einen Schwellenwert (SW) erreicht, wobei die Zeitablauf-Meldeeinheit (11) anschließend den Zähler (10) entsprechend den in dem bereitgestellten Objekt enthaltenen Daten dekrementiert.

(Figur 6)

## Bezugszeichenliste

	1	Datenquelle
	2	Scheduler
5	3	Datenleitungen
	4	Leitungen
	5	Speichersystem
	6	Datenverarbeitungseinheit
	7	Datenbus
10	8	Leitungen
	9	Schreibereinheit
	10	Zähler
	11	Zeitablauf-Messeinheit
	12	Schwellenwerteinstellleitung
15	13	Fehleranzeigeleitung
	14	Dekrementleitung
	15	Zählerstandabfrageleitung
	16	Dekrementleitung
	17	Steuerleitungen
20	18	FIFO-Steuereinheit
	19	Leitungen
	20	Leitungen
	21	FIFO-Speicher
	22	Attributdatenpuffer
25	23	Nutzdatenpuffer
	24	Datenleitungen
	25	Multiplexereingang
	26	Multiplexer
	27	Datenleitung
30	28	Verzögerungsschaltung
	29	Addierereingang
	30	Modulo-M-Addierer
	31	Datenleitung
	32	Eingang
35	33	Effektiver Datenadressgenerator
	34	Zähler
	35	Datenleitung

	36	Eingang
	37	Leitung
	38	Addierereingang
	39	Addiererausgang
5	40	Multiplexereingang
	41	Multiplexer
	42	Überlaufanzeigeleitung
	43	Basis-Adress-Register-Bank
	44	Leitung
10	45	Leitung
	46	Leitung
	47	Verzögerungsschaltung
	48	Datenleitungen
	49	Multiplexereingang
15	50	Link-Adress-Register-Bank
	51	Datenleitungen
	52	Multiplexereingang
	53	Steuereingang
	54	Steuereingang



Scheduler

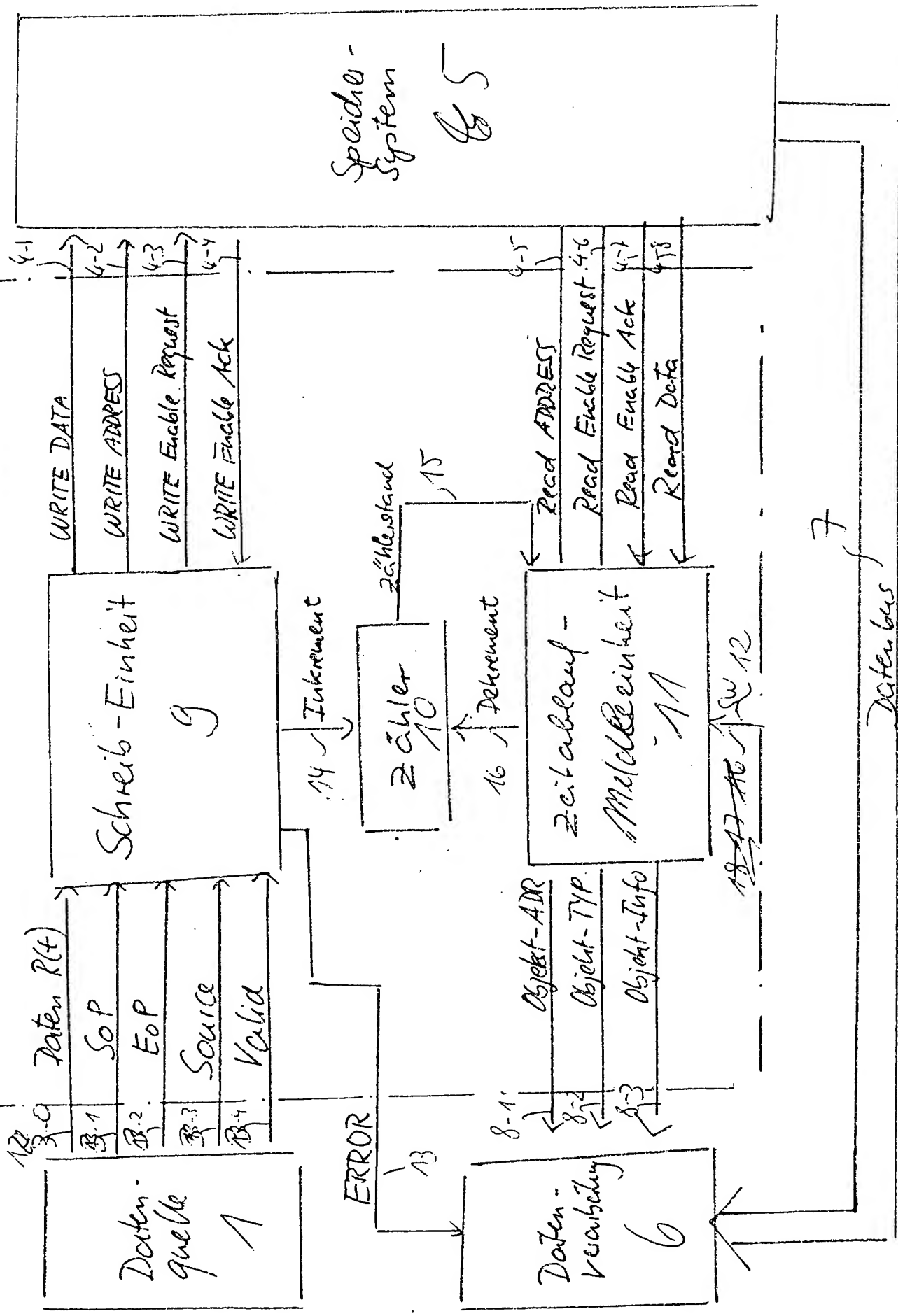
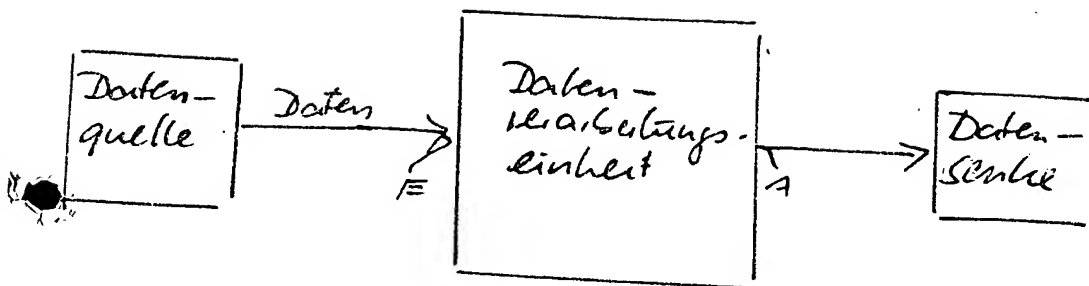


Fig 6

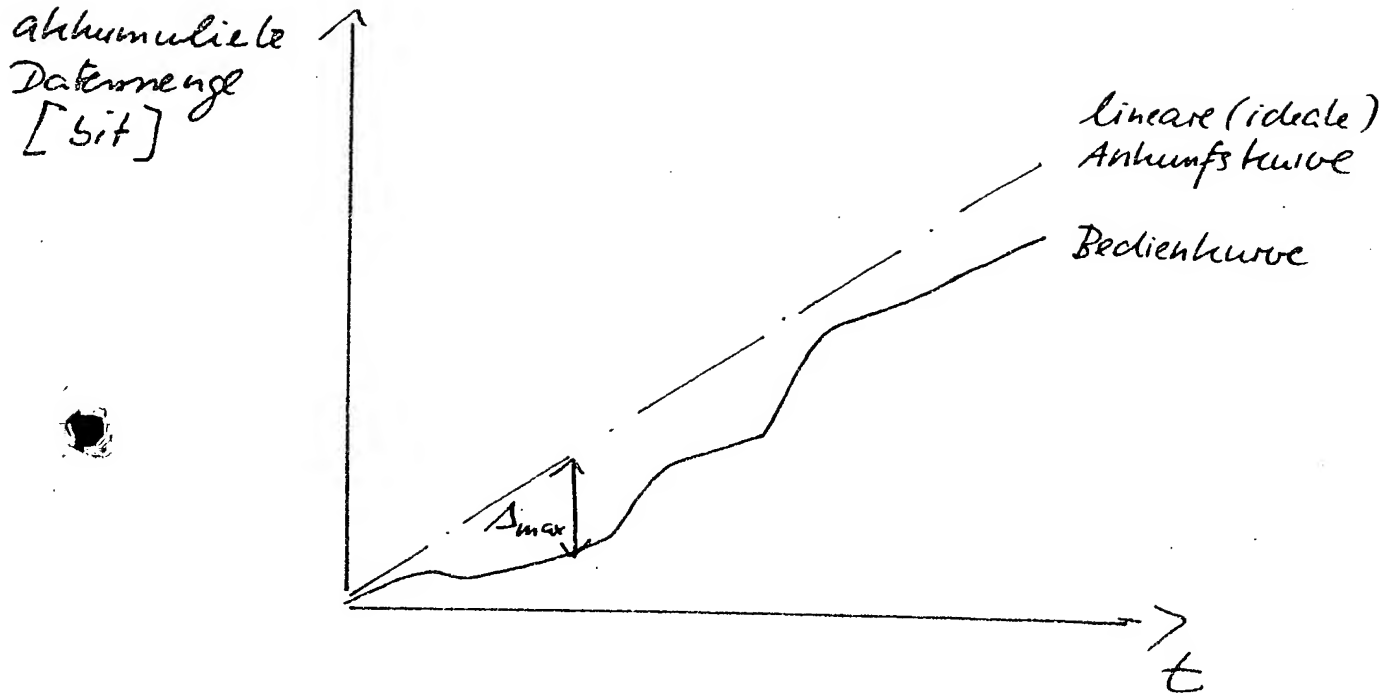
1/16



Skizze der  
Technik

Fig 1

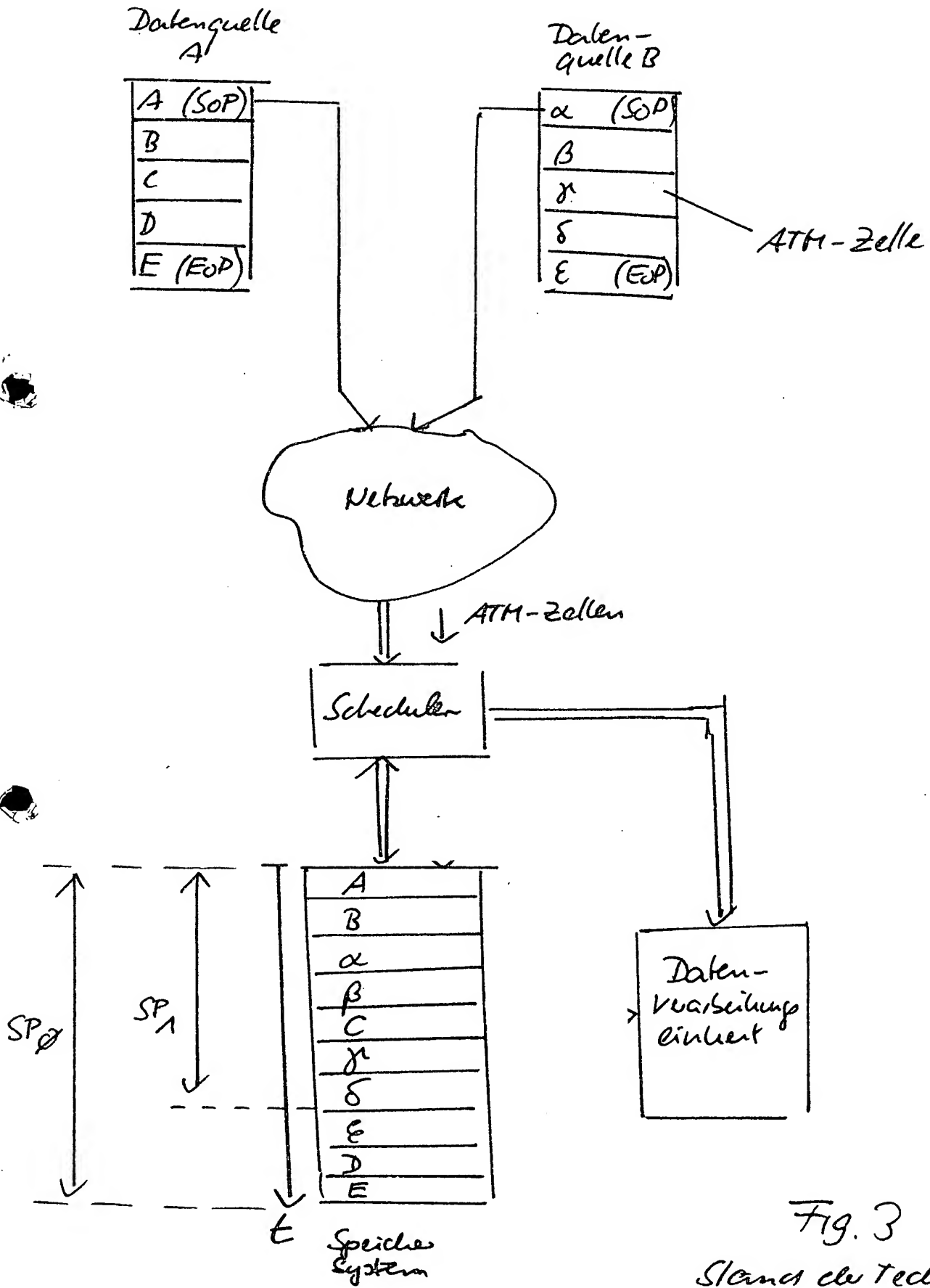
2/10



Stand  
der  
Technik

Fig 2

3/16



4  
16

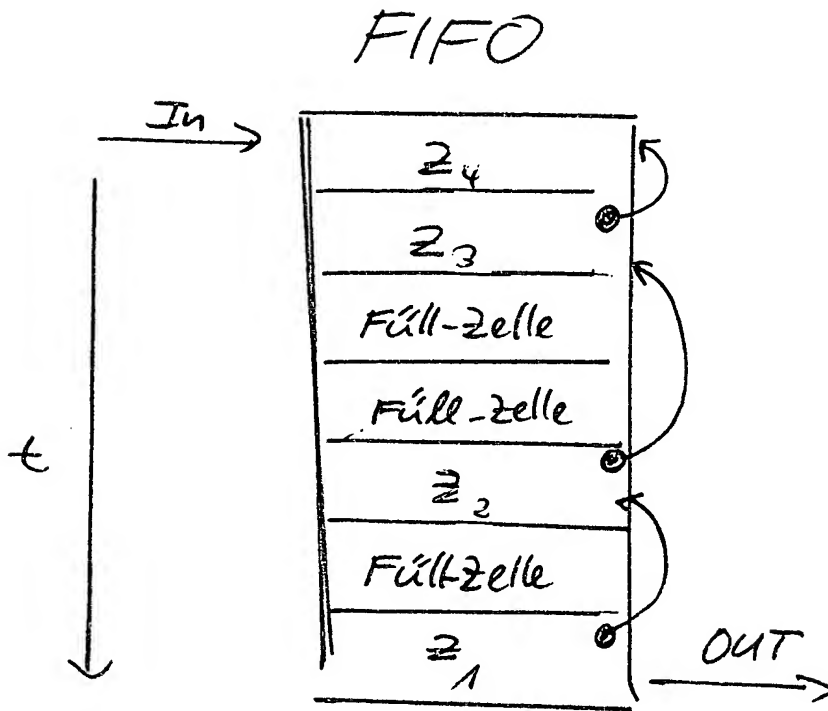


Fig. 4

Stand  
de  
Technik

5/16

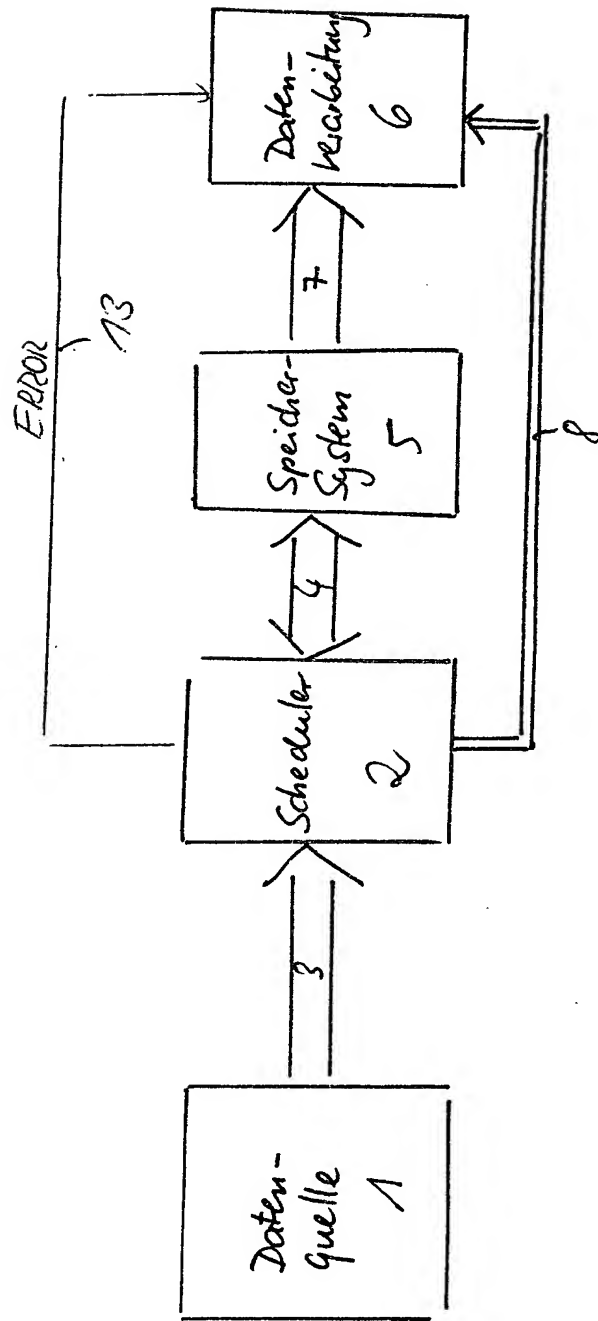


Fig. 5

6/16

# Schedulern

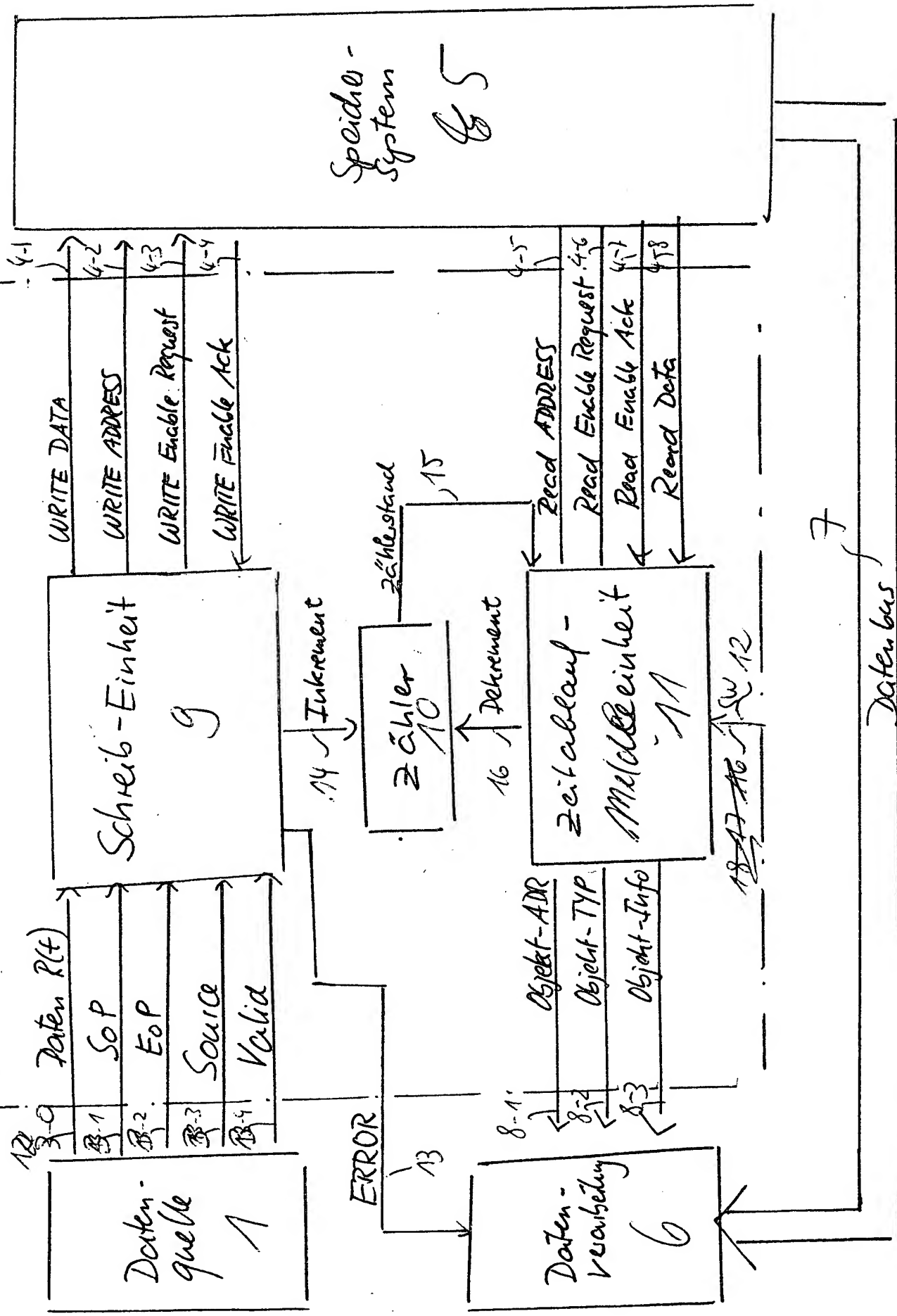


Fig 6

7/16

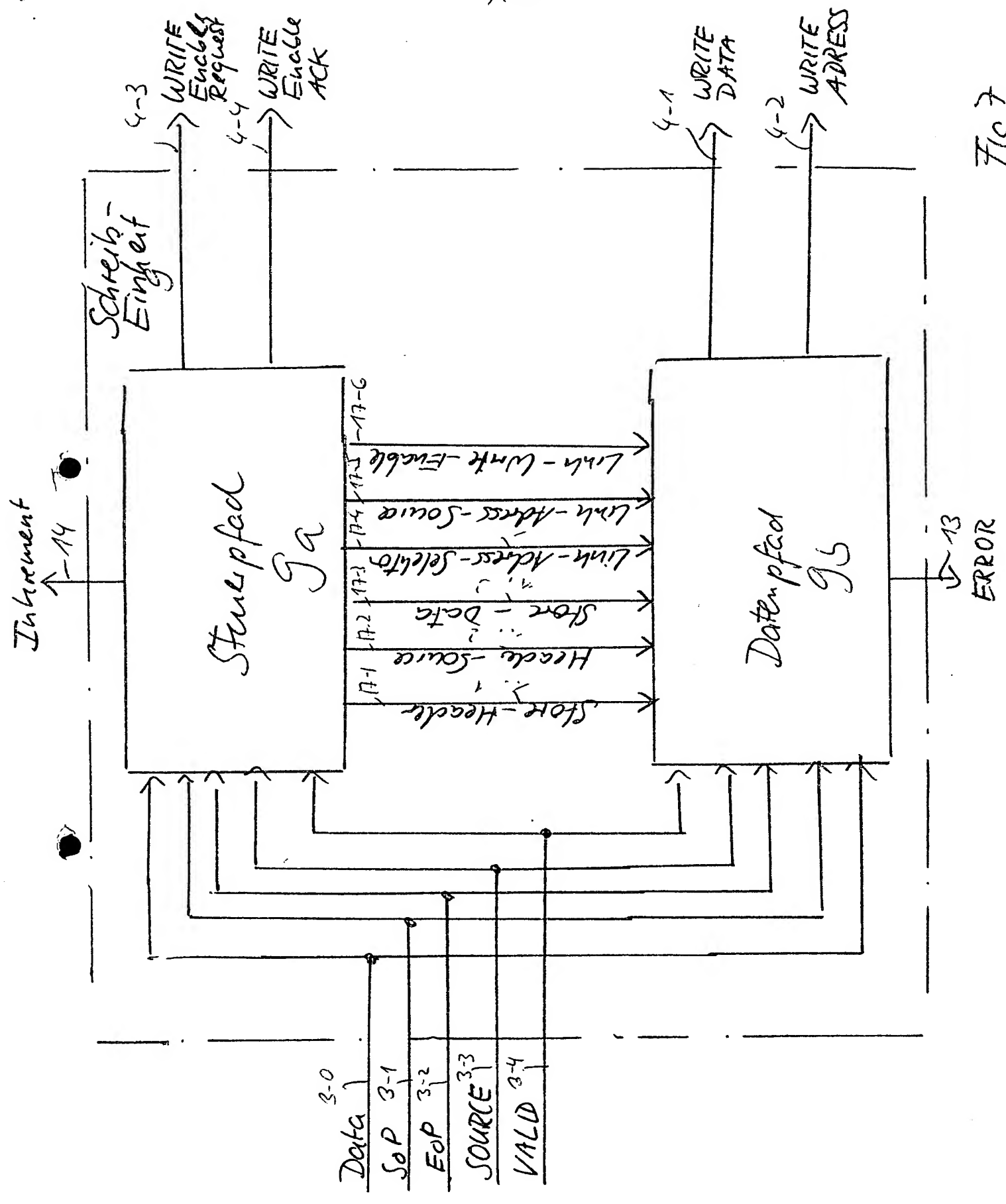
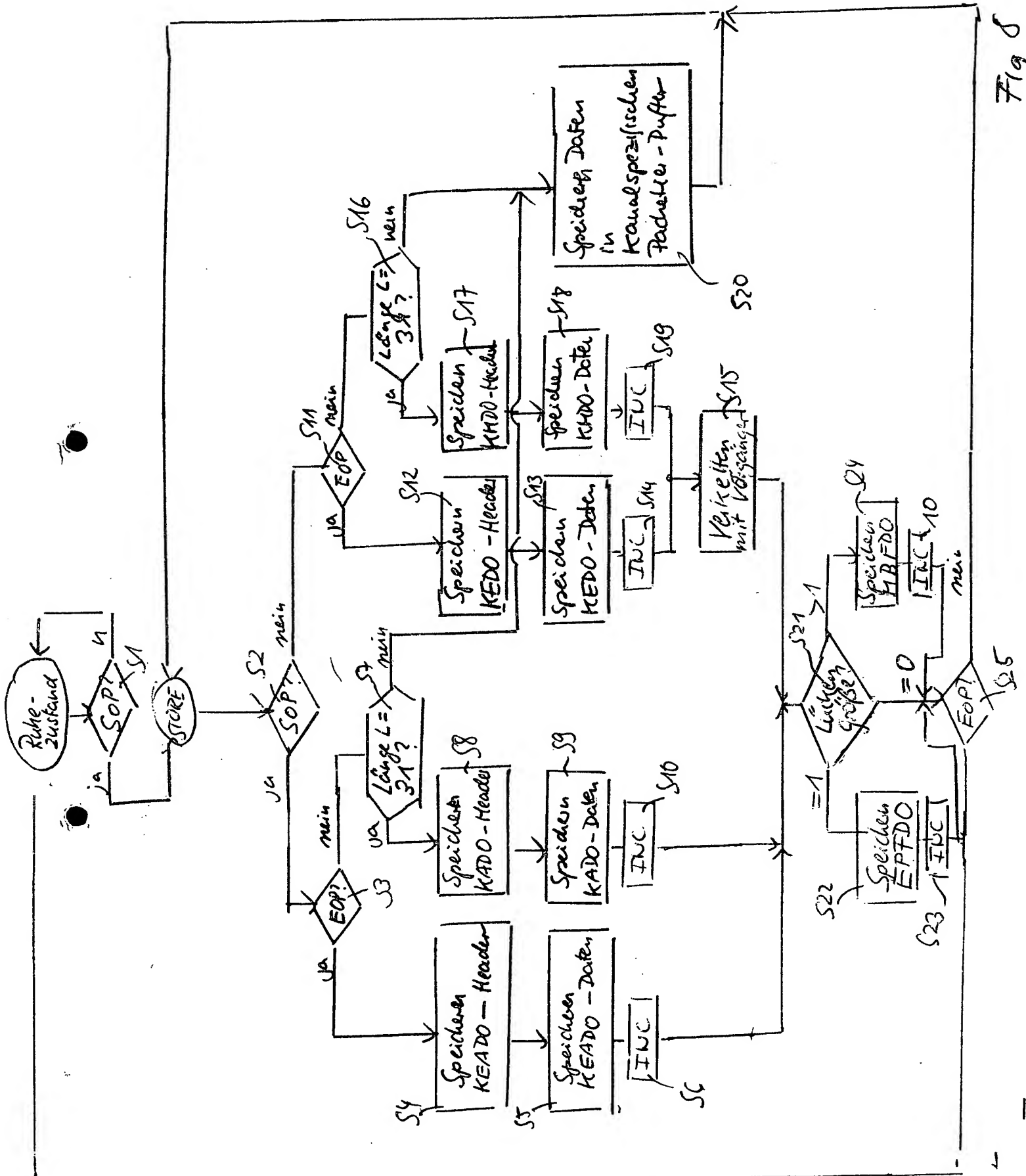


Fig 7





6/8

Datenpaket  
DP

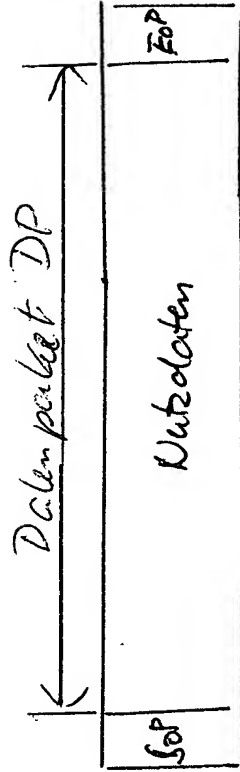
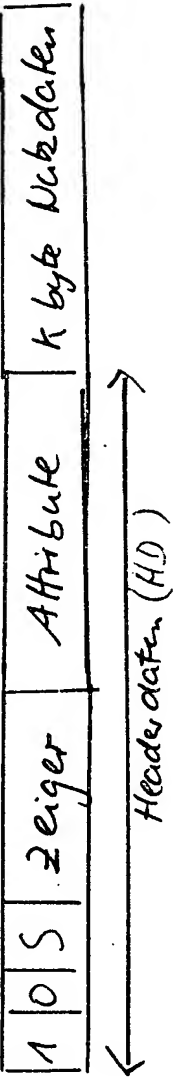
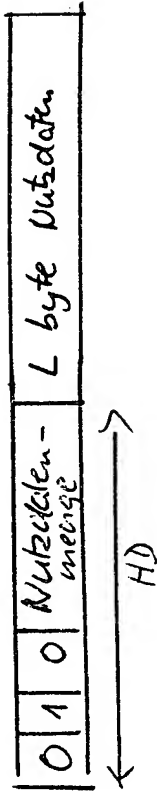


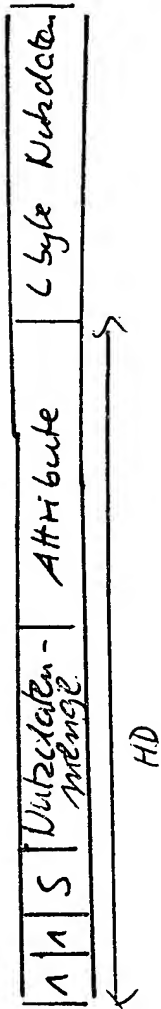
Fig. 9



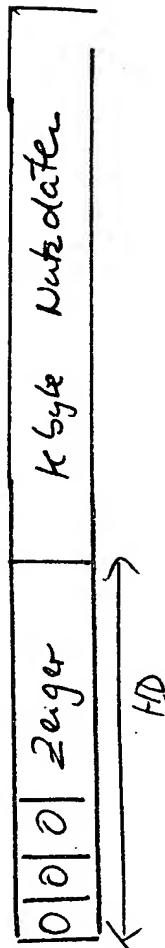
KADO



KEDO



KEADO



KHDO

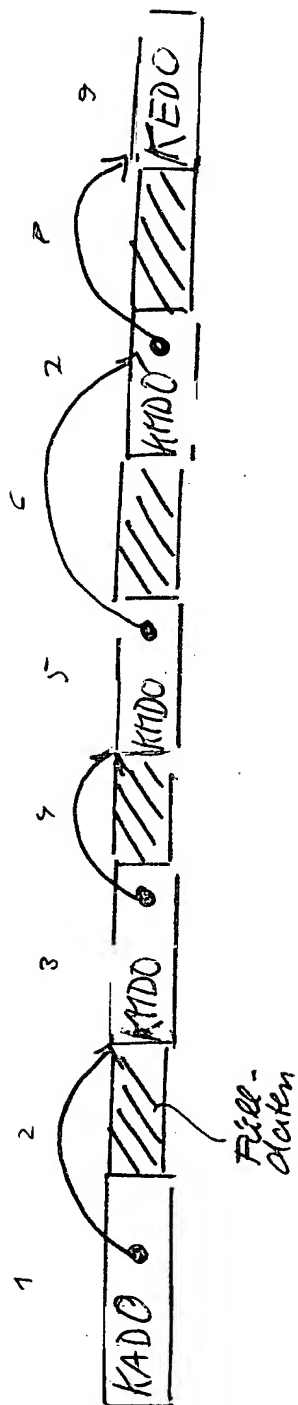


~~EPFO~~  
~~EBFDO~~



~~HPFO~~  
~~HBFDO~~

Fig. 10



M/16

Fig. 11

12/16

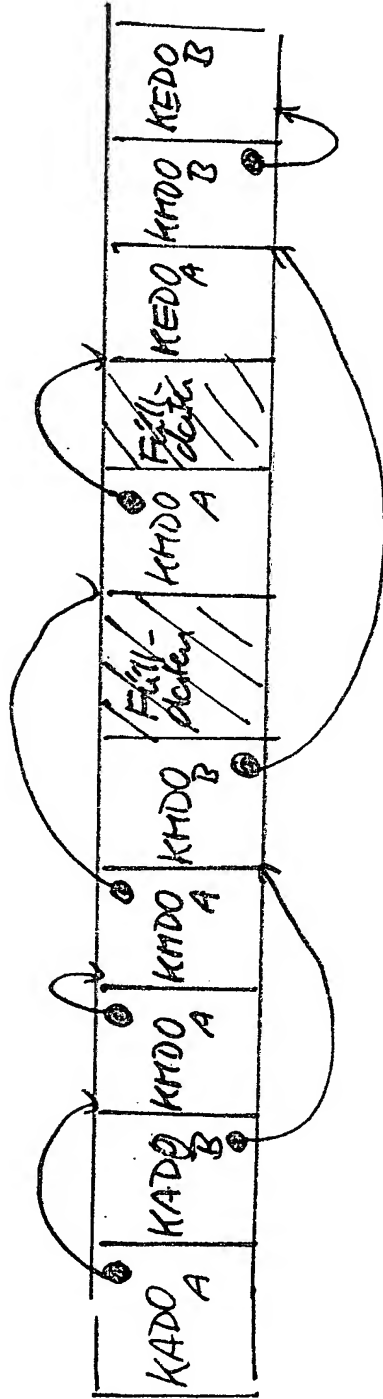


Fig. 12

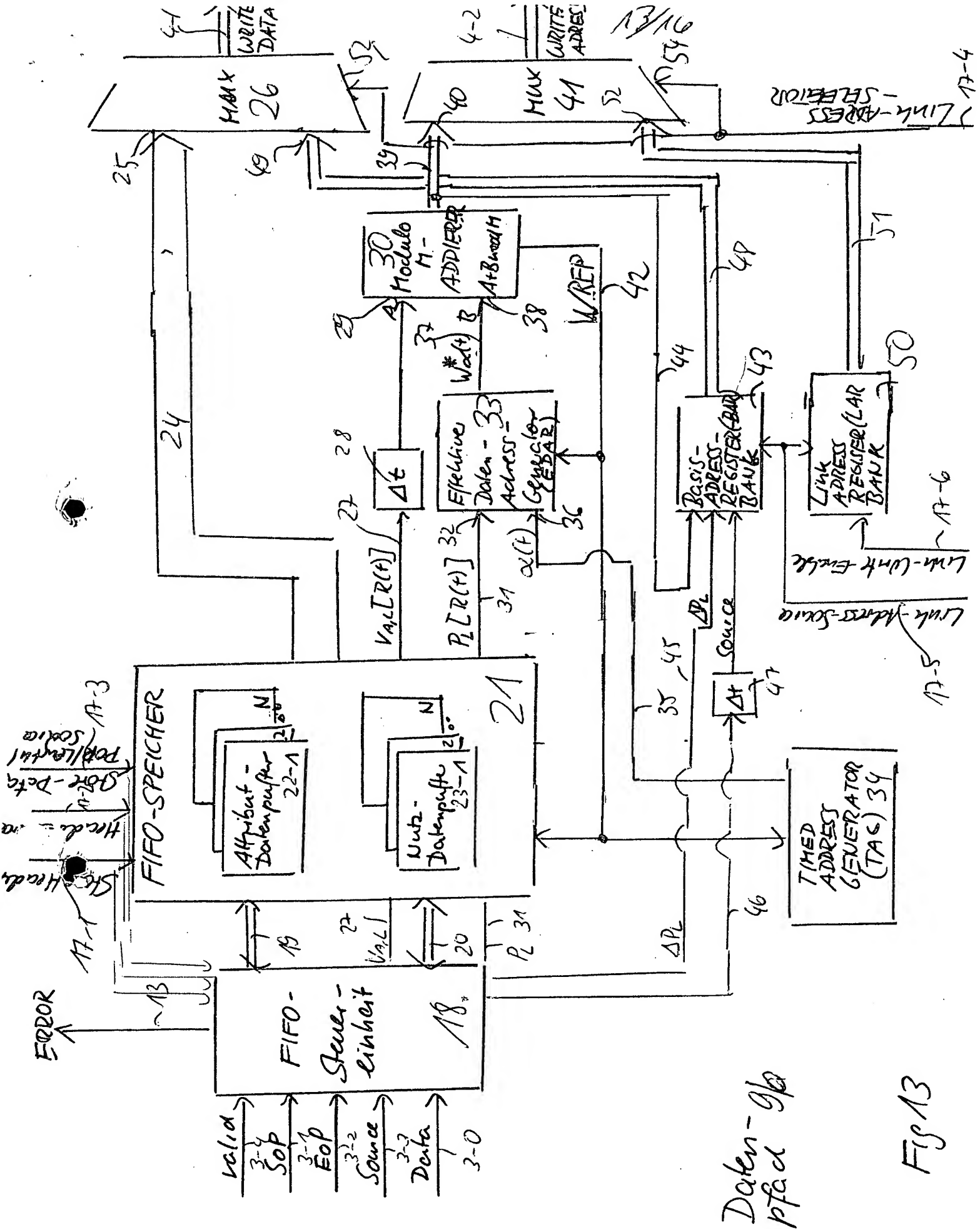


Fig 13

Daten-96  
pfad

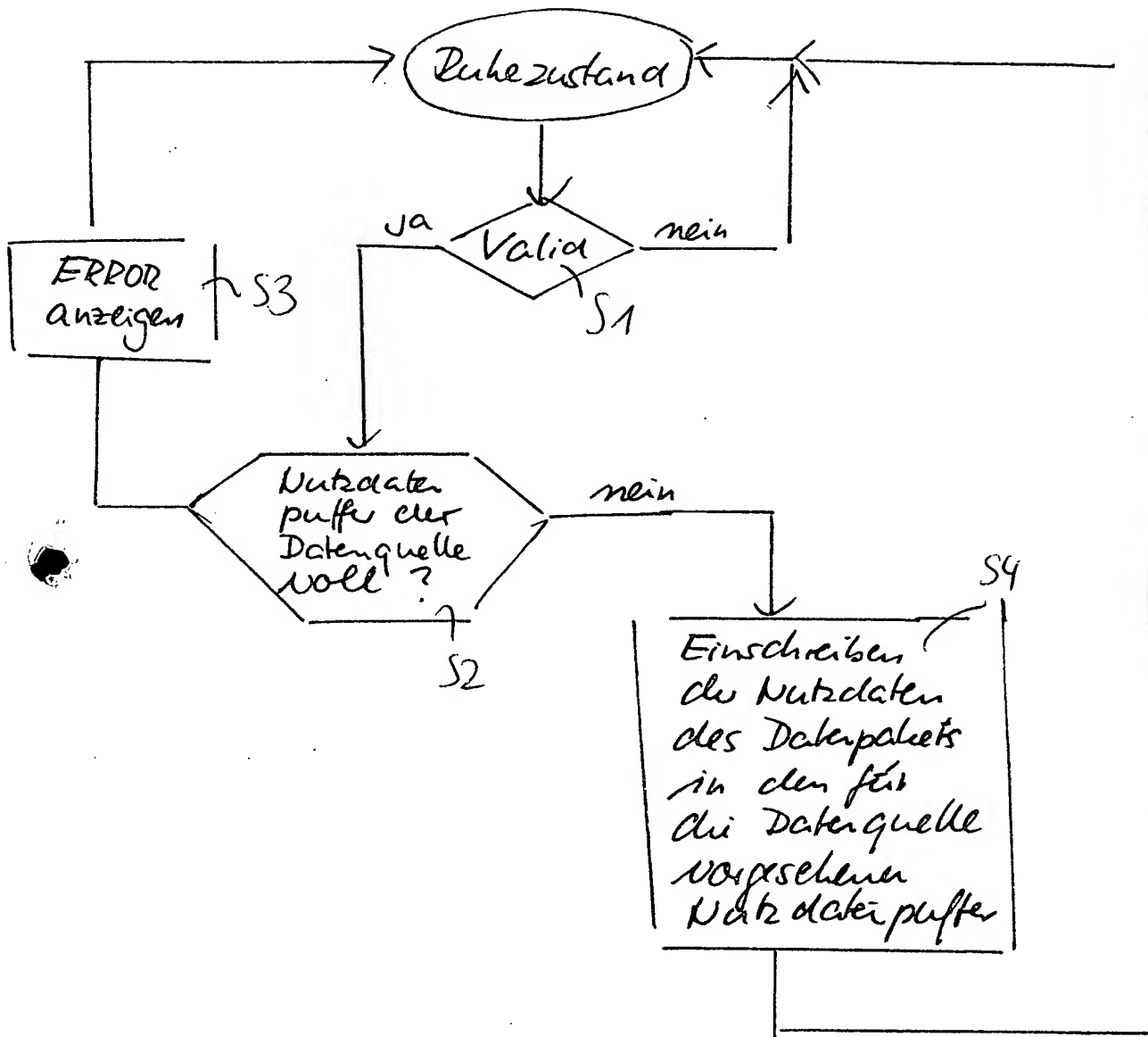


Fig. 14

15/10

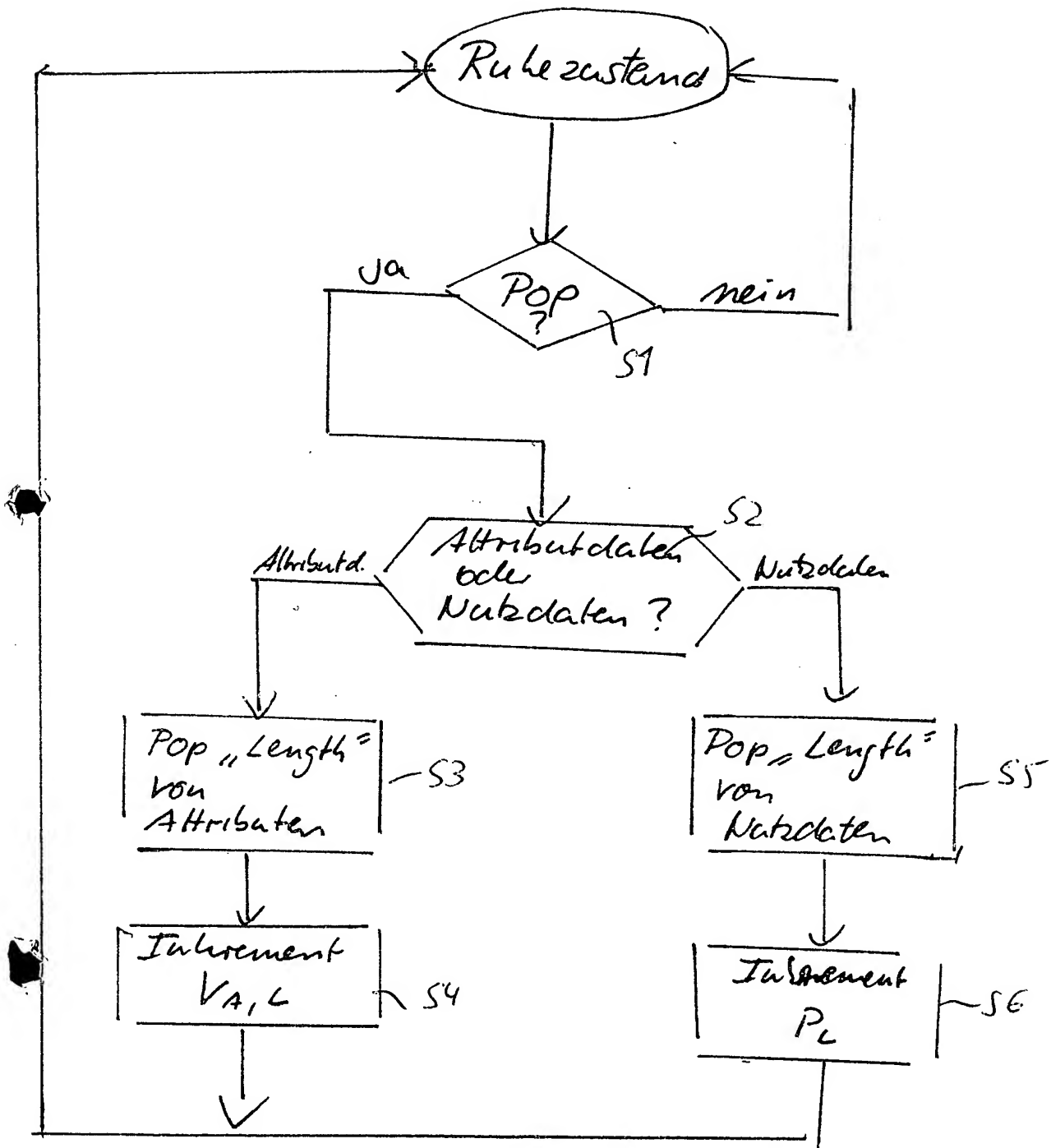


Fig. 15



16/16  
Zertablauf-Meldeinheit

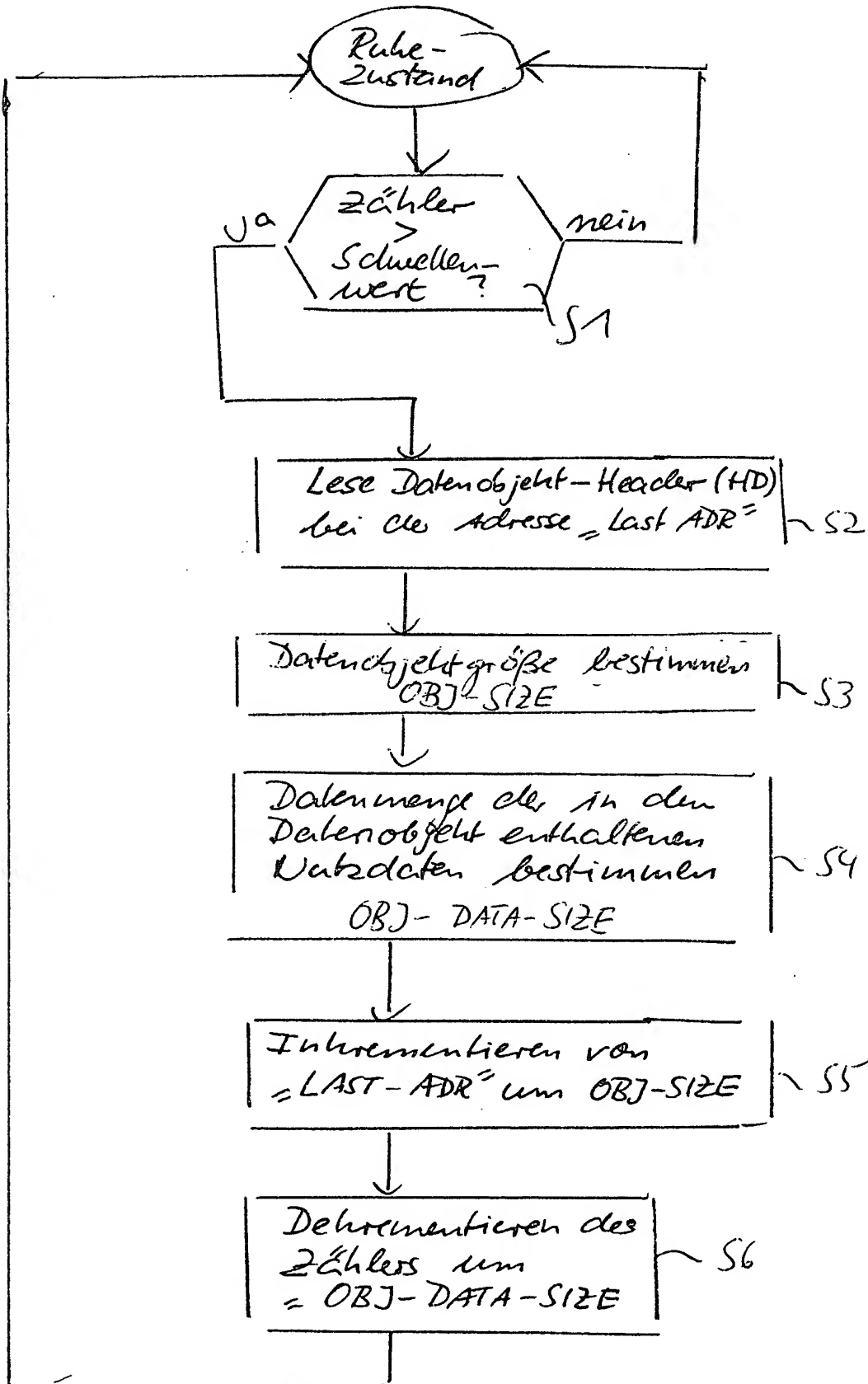


Fig. 16